Answering the Connectionist Challenge:
A *Symbolic* Model Of Learning the Past Tenses Of English Verbs*

Charles X. Ling[†] Marin Marinov[‡]

Abstract

Supporters of eliminative connectionism have argued for a pattern association based explanation of language learning and language processing. They deny that explicit rules and symbolic representations play any role in language processing and cognition in general. Their argument is based to a large extent on two artificial neural network (ANN) models that are claimed to be able to learn the past tenses of English verbs. (Rumelhart and McClelland, 1986; MacWhinney and Leinbach, 1991). In this article we critically review Rumelhart and McClelland's as well as MacWhinney and Leinbach's ANN-models and conclude that they do not succeed in the assigned task of learning the past tenses of English verbs. In order to answer their challenge to the symbolic processing approach, we present our Symbolic Pattern Associator (SPA) – a *general purpose pattern associator* that can learn to associate arbitrary discrete patterns. We carried out several experiments with the SPA using the same set of verbs that was used in MacWhinney and Leinbach's simulation with more realistic training and testing procedures. The SPA outperformed the connectionist models by a wide margin in the accuracy of learning, and successful inductive generalizations to unseen verbs. Our SPA has very natural and psychologically realistic explanations to many psychological effects such as U-shaped learning curve, and is much closer to human subjects in predicting past tenses of the pseudo-verbs. In contrast to ANNs whose internal representations are entirely opaque, the SPA can represent the acquired knowledge in the form of production rules that allow for further higher level processing and integration, resulting in linguistically realistic associative templates for irregular verbs and production rules for regular verbs. In the light of these findings, we conclude that eliminative connectionists' vision of cognition as simple pattern association and pattern recognition *without* symbolic representation is inadequate. Pattern association as such does not imply rule-less or cue-based models of language acquisition or of human learning in general.

---

*Preprint. Appeared in *Cognition*, 49(3), 235–290, December, 1993.

†Department of Computer Science, The University *of* Western Ontario, London, Ontario, Canada N6A 5B7. Email: ling@csd.uwo.ca. Corresponding author.

‡Department of Philosophy, Sofia University, Bul. Tzar Osvoboditel #15, Sofia – 1000, Bulgaria.

MOTTO:

Connectionism has dramatically shifted the mainstream of opinion in cognitive science, but only because the existing implementations actually perform nontrivial tasks in ways unanticipated by recent opinion. If those of the "classical" school want to resume their hegemony, they will need more than a persuasive ideology – they have had that all along – they will need some positive results of actual modelling as striking as those the connectionists have used to attract our attention.

Daniel Dennett

1 Introduction

The current controversy between the supporters of the symbolic and the connectionist approaches in cognitive science has been focused on two main areas of disagreement, roughly following a split inside the connectionist camp between *implementational* connectionism and *eliminative* connectionism (Pinker and Prince, 1988, pp. 76-77). On the one hand, implementational connectionists (Smolensky, 1991b; Touretzky, 1991; Pollack, 1991) agree that parallel distributed processing (PDP) models have to be able to implement symbolic structures like parse trees, part/whole hierarchies, etc. in order to enable them to manipulate mental representations with constituent structure. Such capacities are essential if artificial[1] neural network (ANN)-based cognitive architectures can hope to explain certain pervasive cognitive phenomena like systematicity (cf. Fodor and Pylyshyn, 1988). The disagreement between implementational connectionism and the supporters of the classical (symbolic) approach is centred mainly on the extent to which connectionist architectures can actually implement the classical Turing/von Neumann cognitive architecture and on the consequences of such an implementation for the psychological relevance of implementational connectionism (Fodor and Pylyshyn, 1988; Fodor and McLaughlin, 1990; Smolensky, 1988, 1991a, 1991b).

In contrast, *eliminative connectionism* is a much more radical doctrine. This is the view that "once PDP models are fully developed, they will replace symbol-processing models as explanations of cognitive processes" (Pinker and Prince, 1988, p. 77). According to eliminative connectionists (Rumelhart and McClelland, 1986; McClelland, Rumelhart and Hinton, 1986; Churchland, 1988; 1989) it will be *impossible* to find a principled mapping between the matrixes of connection weights and vectors of activations used in a connectionist ANN, and symbol structures like parse trees, propositions, etc. and the rules for their manipulations used in symbol-processing models based on the Turing/von Neumann cognitive architecture. At the same time, the supporters of this approach believe that the resources of connectionist archi-

[1]We use the qualifier "artificial" following the majority of connectionist researchers who do not insist on the neurophysiological adequacy of their models (Cf. Smolensky, 1988).

tectures will be sufficient to explain in principle all cognitive phenomena without the need to postulate the existence of rules and symbolic representations.

Eliminative connectionism has received a great deal of attention largely due to a famous PDP model developed by Rumelhart and McClelland that was claimed to be capable of learning the past tenses of English verbs without the use of explicit rules and symbolic representations (Rumelhart and McClelland, 1986, pp. 216-271). After several critical reviews of this model by Pinker and Prince (1988), Lachter and Bever (1988) and a detailed background analysis of the model by Plunkett and Marchman (1991), a new revised PDP model was proposed by MacWhinney and Leinbach (1991) (M&L henceforth). They claimed that their new model meets most of the criticisms addressed at the earlier eliminativist model of Rumelhart and McClelland and that it is truly able to learn the past tenses of English verbs without any explicit representation of the acquired linguistic knowledge. MacWhinney and Leinbach also posed a challenge to the symbolic processing approach to cognition to provide a working rule-based learning model that can perform better than their connectionist model.

In this paper we present a *symbolic learning model* which performs significantly better than MacWhinney and Leinbach's model in learning the past tenses of the *same* set of English verbs that was used in their own experiment[2]. Also, our results are far superior than Rumelhart and McClelland's results, even without taking into account the much smaller size of the verb set used by them. In contrast to the eliminative connectionists' claims, we demonstrate that the knowledge acquired by our learning system can be represented in explicit rules. We consider this to be an additional major advantage of our system, since explicit form of knowledge representation is indispensable for a comprehensive account of language learning and language processing. We also show that our symbolic learning model is not application specific and can, in principle, match the learning and representational capabilities of any ANN. These results seriously undermine the claim that connectionist architectures can provide a radical alternative to the classical Turing/von Neumann cognitive architecture.

We also consider the question of whether connectionist or symbolic pattern associators can provide a more psychologically realistic models of language learning. We show that our results are much closer to results obtained from actual psychological experiments than the published connectionist results, and that our symbolic pattern associator can induce rules that predict very well new "unseen" verbs.

[2]Brian MacWhinney (personal correspondence) reports the existence of a new connectionist model that takes into account some of our criticisms. See discussions in Section 6.

2 Review of Previous Connectionist Models

2.1 ANNs as Pattern Associators

A common feature to both Rumelhart and McClelland's and MacWhinney and Leinbach's models is the treatment of the ANN as a device that can learn to associate arbitrary patterns. Both models make much use of the fact that a multilayer ANN using the error-backpropagation algorithm, or a perceptron using the perceptron convergence algorithm can be "conditioned" to associate pairs of arbitrary patterns. In this process, which is essentially a supervised learning from examples, pairs of input and output patterns are provided by the human supervisor or "teacher". A properly structured multilayered ANN can be trained to associate an input pattern with an output pattern using the error backpropagation algorithm. After training, if an ANN model is presented with any of the input patterns it was trained on, it can (in many cases) produce the correct or desired output pattern. Then we can say that the network has learned to associate the two sets of input and output patterns. Moreover, networks trained in such a way appear able to extract (some) regularities in the pairings of input and output patterns and sometimes can respond correctly when presented with input patterns not encountered during training. This suggests that (in some cases) they exhibit a degree of genuine inductive generalization abilities.

According to eliminative connectionism the significance of the ANN as a pattern associator for psychology lies in the fact that on the one hand ANNs can learn to correlate arbitrary patterns and to generalize their knowledge to other "unseen" patterns, but on the other hand, they cannot and need not represent the acquired knowledge in symbolic form; they cannot and need not consult any explicit rules during or after the acquisition of this knowledge. In fact, the "representation" of knowledge in terms of connection strengths and patterns of activations is in numeric form and is entirely opaque to the human interpreter. In this way the ANN as a pattern associator solves the well known problem of knowledge representation by simply dissolving it. Thus, the central thesis of eliminative connectionism is that eventually, with the development of powerful connectionist models, cognition will come to be seen simply as pattern association and pattern recognition. In this way all talk of rules and symbolic representations will be eliminated from cognitive science.

2.2 Rumelhart and McClelland's Model

Rumelhart and McClelland believe that they have made the first step in the gradual elimination of the use of rules and symbolic representations in explaining cognition by developing a PDP model that was credited by them with the ability to learn the past tenses of English verbs. They specifically addressed the issue of the existence of explicit but inaccessible rules in language acquisition associated with the work of N. Chomsky, S. Pinker, and other supporters of the symbol-processing approach. According to Rumelhart and McClelland, their PDP model eliminates the need to postulate the existence of explicit rules in accounting for the human knowledge of language:

> We suggest instead that the implicit knowledge of language may be stored in connections among simple processing units organized into networks. While the behaviour of such networks may be describable (at least approximately) as conforming to some system of rules, we suggest that an account of the fine structure of the phenomena of language use and language acquisition can best be formulated in models that make reference to the characteristics of the underlying networks. (Rumelhart and McClelland, 1987, p. 196)

We will not describe in any great detail Rumelhart and McClelland model since this model is one of the best known connectionist simulations. We will only note that at the heart of their model is a simple perceptron based pattern associator. This pattern associator is connected to an input encoding network and an output decoding/binding network. The encoding network takes as input the phonological representation of the root form of the verbs and converts it into a special Wickelfeature representation format. The Wickelfeature representation of the root form of each verb is then paired with the Wickelfeature representation of its past tense in the perceptron pattern associator for the duration of the training process. The decoding/binding network is used to decode the output of the pattern associator from the Wickelfeature format back into the same phonological format used at the input of the encoding network.

Rumelhart and McClelland claimed that the network after two limited training periods was able to learn correctly the past tenses of both regular and irregular verbs used for training and to generalize this knowledge to previously "unseen" regular and irregular verbs. Moreover, the errors made by the model during the training process, according to them, broadly followed the U-shaped learning curve in the acquisition of the past tense exhibited by young children: early – correct production of regulars and irregulars, medium – incorrectly regularized irregulars, and late correct production of the majority of regulars and irregulars.

According to Rumelhart and McClelland the success of their model confirms one of their main predictions – the nonexistence of explicit rules and symbolic representations mediating language acquisition:

> We have, we believe, provided a distinct alternative to the view that children learn the rules of English past-tense formation in any explicit sense. We have shown that a reasonable account of the acquisition of past tense can be provided without recourse to the notion of a "rule" as anything more than a description of the language. We have shown that, for this case, there is no induction problem. The child need not

> figure out what the rules are, nor even that there are rules. The child need not decide whether a verb is regular or irregular. There is no question as to whether the inflected form should be stored directly in the lexicon or derived from more general principles (Rumelhart and McClelland, 1986, p. 267).

2.3 Criticism of Rumelhart and McClelland's Model

Rumelhart and McClelland's model has received extensive critical attention in the literature. Two of the best known critical reviews were written by Pinker and Prince (1988), and Lachter and Bever (1988). They were followed by more experimental and theoretical work on the learning of the past tenses of English verbs (Plunkett and Marchman, 1991; Kim, Pinker, Prince, and Prasada, 1991; Marcus, Pinker, Ullman, Hollander, Rosen, and Xu, 1993; Pinker and Prince, 1991; Pinker, 1991; Prasada and Pinker, 1993; Egedi and Sproat, 1991). The most important conceptual issues that surfaced during the discussion were connected with the support that the eventual success of Rumelhart and McClelland's model of learning the past tenses of English verbs or similar improved PDP models could lend to eliminative connectionism.

The Induction Problem. Rumelhart and McClelland believe that one of the major results of their simulation is the demonstration that in the learning of the past tenses of English verbs there is no induction problem. But even if their model had a 0% error rate, such a conclusion would not be justifiable. As Pinker and Prince, and Lachter and Bever point out, Rumelhart and McClelland try to model the acquisition of the production of past tense considered in isolation from the rest of the English morphological system. Rumelhart and McClelland assume that the acquisition process establishes a direct mapping from the phonetic representation of the stem to the phonetic representation of the past tense form. (Pinker and Prince, 1988, pp. 87-88) At least two crucial steps are still required to go from the phonetic strings to lexical items and then to morphological categories like "present tense" and "past tense". Without explicit representation of acquired knowledge in phonetic form, it is unclear at all how a PDP model that acts only as a pattern associator, will be able to learn new category information such as "verb stem". Simply remaining at the level of phonetic patterns, it is impossible to express new categorical information necessary for further inductive steps. As Pinker and Prince put it:

> [one of the inherent deficits of the model is that] there is no such thing as a variable for any stem, regardless of its phonetic composition, and hence no way for the model to attain the knowledge that you could add /d/ to a "stem" to get its past (Pinker and Prince, 1988, p. 124).

Representational Format. The representational format used by Rumelhart and McClelland in their model was also subjected to severe criticism. Lachter and Bever pointed out that the results the model actually achieved would have been impossible without the use of several TRICS (The Representations It Crucially Supposes). One such TRIC was reducing the number of Wickelfeatures from about 1000 to 260 not randomly but in such a way as to decrease disproportionally the information provided by some Wickelfeatures. This selective deletion had the effect of reducing the information contained in some Wickelphones and increasing it in the "centrally informative" Wickelphones. (Lachter and Bever, 1988, p. 209) This certainly introduced a bias in the data in favour of a successful outcome. Also, explicitly coding word boundary information in a completely separate set of 200 Wickelfeatures had the effect of giving privileged information status to phonemes in the beginning and at the end of a word. This "boundary sharpening" also introduced a favourable bias (Lachter and Bever, 1988, p. 210). Another criticism is that this format cannot encode all possible words unambiguously. For example, the model would be unable to distinguish between the words *algal* and *algalgal* in the Australian language Oykangand. This means that with this format we cannot represent strings of arbitrary lengths (Pinker and Prince, 1988, p. 97).

Generalization Power. On purely conceptual grounds we are not convinced that Rumelhart and McClelland have provided an adequate model of the learning of the past tenses of English verbs because of the inductive limitations built in their model. But even the performance of the model does not come close to accomplishing the limited task of learning the set of specified mappings between phonetic patterns that Rumelhart and McClelland hoped to achieve.

Rumelhart and McClelland trained the network with 420 regular and irregular verbs. The actual training of the network was carried in two stages, fist it was trained on only 10 high frequency verbs; in the second stage 410 medium frequency verbs were added. At the end of training the network had learned only approximately 98% of the regulars and about 95% of the irregulars (cf. Rumelhart and McClelland p. 246). The testing sample consisted of 86 "unseen" low frequency verbs (14 irregular and 72 regular). One peculiarity of the model was that sometimes it produced no responses at all and sometimes it produced more than one response to a given input stem. If we count all multiple responses that contain at least one error as errors, we can estimate the *maximum* error rate of the model; if we count all multiple responses that contain a correct response as correct, we will have a *minimum* error estimate for the model. The true generalization power of the model under these testing conditions is somewhere between the high error rate estimate and the low error rate estimate.

The results on the testing sample were: 92.9% maximum error rate for the irregulars, i.e. only one out of 14 irregular past tense forms was correctly produced. This was the no change mapping *bid* $\mapsto$ *bid*. But since for two verbs *cling* and *weep*, both a correct and an incorrect past tense candidates were produced, the minimum error estimate for the irregulars is 78.6%. One consolation for this very high error rate might be that most of the incorrect responses on the irregulars were regularization mistakes, like *catch* $\mapsto$ *catched*, or were no change mistakes, like *grind* $\mapsto$ *grind*. The regulars fared better with a maximum error rate of 33.3% and a minimum error rate of 16.7%. The overall error rates for the whole testing sample were 43.1% maximum and 26.8% minimum.

What is particularly puzzling is why for 6 regular verbs the network did not produce any past tense at all, i.e. the response strengths were under the specified threshold. As Pinker and Prince pointed out "This suggests that the reason for the model's muteness is that it failed to learn the relevant transformations; i.e. to generalize appropriately about the regular past." (Pinker and Prince, 1988, p. 124) Even more difficult to explain is a number of "strange" mistakes: e.g. *squat* $\mapsto$ *squakt, mail* $\mapsto$ *membled, tour* $\mapsto$ *toureder, mate* $\mapsto$ *maded, brown* $\mapsto$ *brawned, shape* $\mapsto$ *shipt,* and *sip* $\mapsto$ *sept.* Prasada and Pinker (1993, p. 5) believe that the trained model failed to generalize to the correct past tense forms of these verbs due to the fact that it can represent only the phonological properties of verb stems, not their morphological structure and therefore it systematically errs on irregular sounding verbs derived from nouns.

Training and Testing Procedures. A learning model with a very a high predictive success is not guaranteed to be a psychologically realistic model if its high success rate is achieved at the expense of psychologically unrealistic training and testing procedures. Rumelhart and McClelland trained their network in the first ten epochs on 10 high-frequency verbs, then in another 190 training epochs they presented to the network an additional batch of 410 medium frequency verbs, and finally, they tested the trained network on 86 low-frequency verbs. The idea that children learn first high frequency verbs and then medium and lower frequency verbs is not unreasonable. However, this training procedure is not entirely psychologically realistic. Children never experience such sharp discontinuities in vocabulary size and content (cf. Plunkett and Marchman, 1991, p.47).

One explanation for the adoption of a training procedure which assumes the existence of such a period of "explosive growth" in vocabulary size is that Rumelhart and McClelland wanted to model the well-known phenomenon of U-shaped learning pattern of regular and irregular verbs in children, i.e. the early correct production of a small number of regular and irregular verbs, the intermediate overregularization of irregular verbs (e.g. *come* $\mapsto$ *comed*), and the late correct production of a large number of regular and irregular verbs. However, Marcus et al. (1993) note that during the period of overregularization the proportion of regular verb tokens among all verbs in parental speech to children is only about 20% to 30%; this figure is very different from the 80% proportion of regular verbs chosen by Rumelhart and McClelland for the medium frequency sample. Obviously, the unrealistically large influx of regular verbs in the second training stage has the effect of cancelling the initially acquired tendency in the network to produce irregular patterns. It is unlikely that the network would be able to produce any U-shaped effect if it were trained on a sample with a more realistic proportion between regular and irregular verb tokens. An additional weakness in Rumelhart and McClelland's experiment is that no distinction is drawn between verb types and verb tokens and that they completely ignored token frequency, i.e. they used only a single copy of each verb.

We believe that because of its low predictive success (especially on the regular verbs) and because of the artificial training and testing procedure used in the simulation, Rumelhart and McClelland's model is not a psychologically realistic model of the learning of the past tenses of English verbs. Unfortunately, in the absence of a symbolic learning model that could match even the modest results achieved by Rumelhart and McClelland's connectionist model, eliminative connectionists could dismiss many of the important theoretical objections levied against them by typically claiming that "PDP models may provide more accurate accounts of human performance than models based on a set of rules representing human competence." (McClelland, Rumelhart, and Hinton, 1986, pp. 24-25) In this paper we show that with respect to the learning of the past tenses of English verbs and in many other learning tasks, this need no longer be the case.

2.4 MacWhinney and Leinbach's Model

In a paper recently published in this journal MacWhinney and Leinbach report a new connectionist model of the learning the past tenses of English verbs. They claim that the results from the new simulation are far superior to Rumelhart and McClelland's results and that they can answer all of Pinker and Prince's as well as Lachter and Bever's criticisms addressed at the earlier model.

The major departure from Rumelhart and McClelland's Model is in the change of the representational format. In order to avoid the context-dependence of Wickelphones, instead of coding predecessor and successor phonemes as Wickelphones, MacWhinney and Leinbach use special templates with which to code phonemes and their positions. In their simulation all input verbs were represented in phonetic form with the help of a left-justified and a right-justified template of the form

```
      CCCVVCCCVVCCCVVCCC      VVCCC
left-justified template      right-justified template
```

where C stands for consonant and V for vowel space holders. For example, the verb *bet* is represented in UNIBET coding[3] as /bEt/ and using the template form as

```
bEt         b__E_t____________      _E__t
template:   CCCVVCCCVVCCCVVCCC      VVCCC
            (left-justified)        (right-justified)
```

In this way all English verbs that have more than three syllables, more than three consonantal phonemes in a row, or more than two vocalic phonemes in a row were left out of this

[3]MacWhinney and Leinbach use the UNIBET (MacWhinney, 1990) phoneme coding scheme which allows the assignment of a single-letter ASCII code to each phoneme.

experiment because they did not fit the chosen template. Altogether 2062 regular and irregular English verbs were selected. Of these 10% of the least frequent of the regular and 10% of the least frequent of the irregular were set aside for testing the predictive success of the model and the rest were used for training. However, no results were reported on testing unseen regular verbs. There were 118 irregular past tense forms in the training sample that included some of the most frequent ones.

The actual input to the network was created by coding the individual phonemes as disjoint sets of phonetic features. There were 8 features for vowels (front, centre, back, round, high, middle, low, diphthong) and 10 features (voiced, labial, dental, palatal, velar, nasal, liquid, trill, fricative, interdental) for the consonants. In this way for each vowel slot there are 8 dedicated units of the network and for each consonant slot there are 10 dedicated units, representing the values of different combination of phonetic features. Altogether there are 214 feature/slot input units plus 5 specially dedicated control units which code morphological category information that could be used to switch the response of the network to present, past tense, past participle, present participle, or third person singular. (However, no results were reported other than those for the past tense.) The output layer had only 168 feature/slot output units, reflecting the fact that the output template does not have a right-justified part.

The network had two layers of 200 "hidden" units each fully connected to adjacent layers. This number was arrived at through trial and error. As MacWhinney and Leinbach explain, "the model uses two layers of hidden units, because a model which had only one layer did not do as well at learning the training set." (M&L, p. 143) MacWhinney and Leinbach' s network also had a special set of connections that allowed it to *copy* the left-justified phonological form of the input directly onto the output prior to learning. This feature allowed the creation of a bias in the output nodes that could facilitate the learning process since in most cases present and past forms of English verbs differ slightly. The learning procedure used was the standard form of the error backpropagation algorithm. The network was trained to associate the phonetic patterns representing the present tense of the verbs with the corresponding correct form for the past tense.

2.5 Criticism of MacWhinney and Leinbach's Model

MacWhinney and Leinbach conclude that the performance of their model demonstrates that the ANN has succeeded in its assigned task of learning the English verb paradigm and thus, supported a rule-less cue-based account of the verb inflection acquisition process (M&L, p. 123; p. 151). This claim has to be qualified to apply only to the past tenses of English verbs, because MacWhinney and Leinbach do not report any results on the learning of past participle, present participle, and third person singular. We doubt, however, that even this qualified conclusion is supported by the results from their simulation.

Data Representation. Some of the architectural features of MacWhinney and Leinbach's model are of particular concern. It is possible that by using a template with only 18 (left-justified) consonant/vowel slots a bias can be introduced due to the types of verbs allowed in or rejected from the training and the testing samples. Getting rid of all English verbs that have more than three syllables, more than three consonantal phonemes in a row, or more than two vocalic phonemes in a row because they did not fit the chosen template biases the model in favour of shorter verbs, predominantly of Anglo-Saxon origin against longer verbs, predominantly composite or from Latin and French origin.

Generalization Power. MacWhinney and Leinbach reported that they used 1,650 verbs for training – out of which 119 were irregular. Training the network took 24,000 epochs. By epoch 16,000 all of the regular past tenses of the training sample were learned correctly. However, at the end of epoch 24,000 there were still 11 errors on the irregular pasts. This represents a 9.3% error rate of the irregular verbs used for training. MacWhinney and Leinbach believe that "if we had allowed the network to run for several additional days and given it additional hidden unit resources, we probably could have reached complete convergence." (M&L, p. 151). Surprisingly, despite the importance of test results for assessing the value of a learning model, MacWhinney and Leinbach tested the trained network on only 13 unseen irregular verbs. The result was that "9 of these untrained past tense irregulars were missed." (M&L, p. 146) This represents 69.2 error rate on the irregulars. However, for no obvious reason they did not test their model on any of the unseen regular verbs: "Unfortunately, we did not test a similar set of 13 regulars." (M&L, p. 151)

There are serious difficulties in estimating the significance of these results. Because the testing sample is so small – only 11% of the irregular verbs in the training sample and a mere 0.8% of the whole training sample, and because we do not have any results on the unseen regulars, we are entirely in the dark regarding the overall predictive success of the model. Thus, despite the reported results we still do not know anything truly significant about the inductive capabilities of MacWhinney and Leinbach's model. A model without a clearly assigned predictive accuracy is as good as a model with 0% predictive accuracy. Guessing correctly 4 out of 13 test verbs cannot even begin to reveal the realistic error (success) rate of the model because of the unreliability of the testing sample which was used to measure it.

MacWhinney and Leinbach appear to be unconcerned with their choice of an unreliable testing sample and with the low generalization capacity of the model reported on the basis of this sample. Their discussion of the achievements of their model on pp. 146-53 is based mainly on results obtained from the training process. They seem to believe that simply because the network was *trained* to associate certain phonetic patterns, the network has *learned* to produce the past tenses of English verbs. For example they write:

> Performance on the regular past tense was perfected about midway through the

simulation. By the first test point [i.e. the first check during the training process on the training verbs] it was already over 99% correct. In overall terms, the simulation did very well at its task of learning the past tense (M&L, p. 146).

MacWhinney has taken into consideration our criticisms regarding the need for better testing of the generalization capacities of learning models. He reports the existence of a new connectionist model with better generalization capacities than MacWhinney and Leinbach's 1991 model. See Section 6 for more details.

Training and Testing Procedures. In comparison with Rumelhart and McClelland's model, MacWhinney and Leinbach *train* their model in a more realistic setting. They use multiple verbs tokens which enables them to present high frequency verbs approximately every epoch throughout the 24,000 training epochs, and the low frequency verbs are presented approximately once every 700 epochs. This appears to be a more natural way of presenting the data to the model but is still only a *very rough approximation* to the natural learning environment. However, the model *failed* to exhibit the U-shaped learning curve in learning irregular verbs reported by Rumelhart and McClelland in their simulation. MacWhinney and Leinbach reported only a linear improvement of the performance of their model on the irregular verbs. This means that they could not model the first phase of (predominantly correct) production of irregular verbs in children. Unfortunately, since they were unable to trace the rate of learning for the regular verbs, there is no basis for deciding whether the model had exhibited any of the latter portions of the U-shaped learning curve, or simply gradually improved its performance on both regular and irregular verbs. This negative result on the U-shaped test lends additional support to the conjecture that the U-shaped learning curve of Rumelhart and McClelland's model is an artifact of the uneven supply of irregular and regular verbs in the first 10 and in the subsequent 190 epochs of training of their model.

Our analysis of performance of MacWhinney and Leinbach's model leads us to the conclusion that they have failed to provide a psychologically realistic model for learning the past tenses of English verbs. There is no indication that they even produced any results for past participle, present participle, and 3d person singular, so their claim that the model has succeeded in learning the English verb paradigm is unfounded. (Here again MacWhinney and Leinbach identify the training of the network with successful learning.) In many respects MacWhinney and Leinbach's 1991 model does not represent an improvement over Rumelhart and McClelland's model and in some respects, e.g. in overall error rate and in the openness of the reporting of the errors their model made, it actually represents a step back. It must be rejected as a model of the competence of native speakers to learn the past tenses of English verbs and it can hardly serve as an example of how to eliminate rules and symbolic representations from the explanation of the human knowledge of language.

2.6 Is There a Better Symbolic Model?

We entirely agree with MacWhinney and Leinbach when they say that conceptualizations (theories) should ultimately be accepted or rejected in view of the success or failure of their implementations (experimental predictions). We entirely agree with them when they say in a section titled "Is there a better symbolic model?" that

> If there were some other approach that provided an even more accurate characterization of the learning process, we might still be forced to reject the connectionist approach, despite its successes. The proper way of debating conceptualizations is by contrasting competitive implementations. To do this in the present case, we would need a symbolic implementation that could be contrasted with the current implementation (M&L, p. 153).

We decided to take up MacWhinney and Leinbach's challenge and to provide a symbolic learning model that can learn the past tenses of English verbs as well as or even better than any PDP model so far. We hope that the success of our symbolic learning model will lead eventually to the rejection of the eliminativist approach to language learning and language processing in general.

3 The Symbolic Pattern Associator

The core of our model is a device that we call the Symbolic Pattern Associator (SPA for short). The development of the SPA is a result of our search for a general purpose symbolic system that will be able to compete directly with connectionist models on many tasks traditionally considered to be learnable only by ANNs. In order to demonstrate that a symbolic learning model can successfully answer the connectionist challenge, our SPA has to meet two conditions:

- Like ANNs, the SPA must be able to learn to associate sets of arbitrary patterns and it must be able to generalize to new patterns.
- Unlike ANNs, the SPA must be able to represent the knowledge acquired during the learning process in an explicit form.

The SPA is based on the well-known decision tree learning algorithm ID3[4].

[4]Several modifications in the C4.5 program are made. See (Ling & Marinov, 1993).

3.1 The ID3 Symbolic Learning System

Many connectionists believe that neural networks possess some unique advantage over symbolic systems in that they are capable of learning things that symbolic systems are inherently incapable of. It is also quite common to read that symbolic systems are hard and brittle while connectionists systems are soft and flexible and can account for a wider range of cognitive phenomena (Smolensky, 1988). Such beliefs are widespread because many people are not aware of the fact that there are several symbolic learning algorithms that can compete quite successfully with ANNs on a wide range of practical learning tasks and even with the most influential connectionist simulations like NETtalk (Atlas et al., 1990; Shavlik, Mooney, and Towell, 1991; Marinov, forthcoming). One of the most widely studied of these systems is ID3 (Quinlan, 1986a).

The ID3 learning system has been designed by Quinlan (1986a). C4.5 is a latter improved implementation of ID3 (Quinlan, 1986b, 1989). ID3 is a descendant from Hunt, Marin and Stone's CLS (Hunt, Marin, and Stone, 1966) and falls into the class of TDIDT (top-down induction of decision trees) learning systems. ID3 is suitable for a comparison with the back-propagation (BP) algorithm (ANNs using BP) because both ID3 and ANNs using BP are members of the class of supervised learning systems that use propositional representation. In addition, neither of them is application specific. Examples of successful applications for ID3 are chess strategies, weather prediction, medical diagnosis, voting pattern predictions, credit card application assessment, etc. ANNs with BP have also been used in a wide array of applications.

The difference between ID3 and backpropagation lies in the form of knowledge representation:

- ID3 represents the acquired knowledge as decision trees labeled by attribute values and class values. Thus, a decision tree can easily be converted into a set of production rules. In many cases this form of knowledge representation is "transparent" to the human observer i.e. humans are able to understand and use these rules.
- ANNs using BP store the acquired knowledge in their weights or connection strengths in a numerical format. The knowledge stored in this way is completely "opaque" to the human observer.

To understand how the decision-tree building algorithm of ID3 works, it is useful to look at a concrete simulation. ID3 is capable of supervised learning from examples, so ideally it should be able to acquire expert knowledge by learning from examples of human expert decisions. Consider the following case: Many newspaper and radio stations have experts who check the weather forecast for the day, the current temperature and humidity, and give a simple piece of advice: whether the day will be suitable, or unsuitable, for playing golf. ID3 can be trained by providing it with a sample of the past decisions of such an expert.

In order to be trained, ID3 is given examples from the same universe of objects that are taken into consideration by the human expert. For this learning task, the universe will be the days of the week. Each day can be described with the help of the following attributes:

- outlook: with values = {sunny, overcast, rain}.
- temperature: with values = {cool, mild, hot}.
- humidity: with values = {high, normal}.
- windy: with values = {true, false}.

Given such a description the expert has to decide whether the day will be suitable for playing golf or not. The task is to classify each object in the universe as belonging to one or other of two classes which we may call P (play golf) and N (do not play golf).

Suppose we are to provide ID3 with the set of training examples in Table 1, say of Saturday mornings in the past few years together with the decisions of a human expert. Each object in the universe can be encoded as a four dimensional state vector, where each vector component v_i of $\mathbf{v} = [v_1, v_2, v_3, v_4]$ corresponds to the value of a different attribute. ID3 builds a decision tree that will effectively classify all examples (vectors) in the training set, then this tree can be used to predict future instances not included in the training set. ID3 builds decision trees by employing a simple divide and conquer strategy. This strategy is recursively applied in building subtrees, until all remaining examples in the training set belong to a single concept (class); then the terminal leaf (circles in the figures of decision trees) is labeled as that concept.

The divide and conquer strategy works as follows: ID3 first picks an attribute based on the information gain of the attributes (see below) – in this case the attribute outlook – and checks its values. As it turns out, all objects with value overcast for the attribute outlook belong to the play golf class, so ID3 closes this branch and assign P to a leaf. In case a value does not classify the subset of examples falling into this branch as belonging to a single class, the same procedure is applied to the subset recursively, looking at another attribute and its values and building subtree for the subset. The algorithm halts when all subsets of objects are classified into their corresponding single classes. In this way ID3 builds the following simple decision tree to classify all of the training examples in Table 1.

But how does ID3 choose from which attribute to start building the subtree? For example, if ID3 chose the first node to be the attribute humidity, a larger and much more complex decision tree would be constructed. ID3 uses a heuristic principle similar to Occam's razor: it uses an expected information gain criterion to choose the *most discriminative* or *most relevant* attribute as the root of a (sub)tree, i.e. the attribute that reduces randomness in the remaining data as much as possible. For a set A of k examples, if p_1 of them belong to the class P, and p_2 of them

Attributes				Class
Outlook	Temperature	Humidity	Windy	
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Table 1: A small training set.

belong to the class N ($p_1 + p_2 = k$), then the *entropy* (i.e. randomness) of the set is:

$$E(A) = -\frac{p_1}{k}\log_2\frac{p_1}{k} - \frac{p_2}{k}\log_2\frac{p_2}{k}$$

Clearly $E(A)$ reaches the minimum value 0 when $p_1 = k$ or $p_2 = k$ (i.e. thiis is a single class and therefore no more classification is needed). Likewise, $E(A)$ reaches the maximum value 1 when $p_1 = p_2 = k/2$ (i.e. one cannot make any better prediction than the 50/50 chance).

If an attribute X has values $x_1, x_2, ...x_v$ and is chosen as a root of the (sub)tree, the set A will be split into v subsets $B_1, B_2, ..., B_v$ according to the values of the attribute X. The entropy of the resulting decision tree $E(X)$ (with X as a root) is a weighted sum of the entropies of v subsets. That is:

$$E(X) = \frac{|B_1|}{k}E(B_1) + ... + \frac{|B_v|}{k}E(B_v)$$

To select the most informative attribute, ID3 chooses a new attribute X such that the information gain $E(A) - E(X)$ is maximized among the remaining attributes. This results in trees that generally branch out from the more informative to the less informative attributes, thus simplifying the overall tree structure. Since the heuristic is applied at one level at time, it does not guarantee to yield the smallest decision tree. However, the use of this heuristic results in building *small* decision trees which in general have a good generalization power. Also, ID3 scales up exceptionally well; it is polynomial with respect to the total number of attributes, the number of attribute values, and total tree size.

It is easy to see that the decision tree beside Table 1 can immediately be converted into a set of production rules just by tracing all of its branches from the root to the terminal leafs – the terminal leaf is the consequent and the conjunction of all roots is the body of the rule.

These rules produced from the tree beside Table 1 have the same predictive accuracy as the decision tree itself:
Rule 1: If outlook = overcast, then play golf.
Rule 2: If outlook = sunny and humidity = normal, then play golf.
Rule 3: If outlook = rain and windy = false, then play golf.
Rule 4: If outlook = sunny and humidity = high, then do not play golf.
Rule 5: If outlook = rain and windy = true, then do not play golf.

3.2 Previous Experimental Comparisons between ID3 and ANNs

We saw that ID3 and the other TDIDT algorithms are very well suited for pattern classification tasks. The ANNs are traditionally believed to be extremely good algorithms for such tasks, especially with noisy and incomplete data. However, ID3 has been improved (such as in C4.5) to deal with noisy and incomplete data as well. So it is important to compare the performance of ID3 and ANNs on real-world data in order to find out if any of these learning systems has a particular advantage over the other.

Many very interesting experimental comparisons between ANNs using BP and the ID3/C4.5 learning algorithm (and other symbolic, statistical and neural network algorithms) have been carried out on the the real world data sets and a few artificially created data sets. Most of the data sets were noisy; and several papers explored the effect of increasing degeneration of the data sets to the degeneration of performances of the learning algorithms. Some small scale comparisons involving a few learning algorithms and/or a few selected data sets have been done (Tsaptsinos et al., 1990; Gorman and Sejnowski, 1988; Thrun et al., 1991; Atlas et al. 1990), but the results were inconclusive since the data sets were too small and selective, or little analysis was done (Thrun et al., 1991). Several more extensive studies have been published, such as (Weiss and Kulikowski, 1991; Fisher and McKusick, 1989; Shavlik, Mooney, and Towell, 1991; Feng et al., 1992; Ripley, 1992). Shavlik, Mooney and Towell (1991) carried out a thorough experimental comparison between ID3, ANNs using BP and the perceptron on five real-world data sets including the famous connectionist text-to-speech conversion model NETtalk (Sejnowski and Rosenberg, 1987). In the StatLog project (Feng et al., 1992), 18 different symbolic, statistical and neural network learning algorithms were critically (and more or less fairly) compared over 6 large data sets from industrial applications. Most of these extensive experimental comparisons have some similar findings:

- No single method (symbolic, statistical, or neural network) performs better than all other methods for all data sets.
- ANNs using BP constantly produce reasonable, near best accuracy, while symbolic methods sometimes performs very well (i.e. outperforms BP) but sometimes not so well.

- The learned results from symbolic learning algorithms are much easier to understand and interpret than those from neural network learning algorithms. This issue is very important if the acquired knowledge are to be useful in most critical decision making, diagnosis, knowledge discovery, and integration of several learning systems. ANNs usually produce little insight on what is learned and why certain solutions are obtained.

- Symbolic/statistical methods are much easier to set up and use than neural network algorithms, which require considerable amount of tweakings of the parameters to make it work well (such as number of hidden units, learning rate, temperature, and normalisation method).

- BP is usually several orders of magnitude slower than symbolic (and statistical) methods. Note that it is unclear if the difference in the running time between the BP and symbolic methods will be reduced by using parallel hardware since most symbolic systems can be implemented in parallel with significant gains in speed as well.

Further research is needed in formal characterization of problems and data sets that can be best solved by each learning method. This is also one of our current research subjects.

3.3 The Architecture of the Symbolic Pattern Associator

Although, ID3 and C4.5 as well as other TDIDT symbolic algorithms have been very successful in challenging the learning and the representational capabilities of neural networks they are essentially limited in the type of learning tasks they can perform. If the learning task is to learn to map a set of different patterns (vectors) into several mutually exclusive classes, as many of the most successful connectionist simulations actually do, ID3 and C4.5 have been shown to perform as well as, or even better than neural networks. However, if the task is to associate a set of patterns with another set of possibly partially overlapping patterns, the neural networks retain a distinctive advantage. The reason is that these are typical *pattern association* tasks, to which ID3 and C4.5 as *exclusive classifiers* are not particularly well suited. For example, if the patterns listed in the right hand side of Table 2 are to be associated, and if ID3 is applied directly, it will treat the output patterns as mutually exclusive classes. As a result, there will be a total of $2^5 = 32$ classes! But then, by using ID3 we will lose the chance to capture any generalizations *within the output patters*, i.e. we will not be able to capture the similarity between the input and the output patterns in Table 2.

In order to rival fully the learning capabilities of ANNs we have to increase the powers of ID3, C4.5, and similar TDIDT systems and to turn them into general purpose symbolic pattern associators. We did that by combining the power of individual decision trees into a joint set of trees all of which work together to accomplish the task of associating two (sets) of arbitrary patterns. Each tree takes as input the same set of input patterns that is available to all other

ID3		SPA	
Input Patterns	Class	Input Patterns	Output Patterns
1 0 0 0	1	1 0 0 0	1 1 0 0 0
0 1 0 0	0	0 1 0 0	0 1 1 0 0
0 0 1 0	0	0 0 1 0	0 0 1 1 0
0 0 0 1	1	0 0 0 1	0 0 0 1 1

Table 2: Pattern classification vs pattern association

trees but is concerned with determining only a portion of the output pattern – usually the value of one of its attributes. More specifically, if the input attributes ($\mathbf{I}_1$ to $\mathbf{I}_n$) and the output attributes ($\mathbf{O}_1$ to $\mathbf{O}_N$) of a pair of input/output patterns can be represented as:

$$\mathbf{I}_1, ..., \mathbf{I}_n; \quad \mathbf{O}_1, ..., \mathbf{O}_N$$

then a total of N decision trees are built, one for each output attribute $\mathbf{O}_i$ $(1 \leq i \leq N)$ taking all input attributes $\mathbf{I}_1, ..., \mathbf{I}_n$ per tree. Once all of N trees are built using ID3, they can be used jointly to determine any output pattern $\mathbf{O}_1, ..., \mathbf{O}_N$, given any input pattern $\mathbf{I}_1, ..., \mathbf{I}_n$. (See Figure 1.) Clearly, any pattern association tasks can be solved by the SPA. Similar to ID3, our SPA is very general and simple to use. The set of trees associating the two sets of patterns can be build serially or in parallel since each tree is built independently from the others. A parallel implementation can lead to great gains in speed.

Figure 1: SPA: A joint decision method for associating arbitrary patterns.

What is very interesting, however, is that once the SPA has been trained, i.e. once the set of associating trees has been built, it can use the decision tree acquired during the training process to generalize from past cases to unknown future cases. In effect, the SPA can rival the learning capabilities of any artificial neural network using a supervised learning algorithm that can learn to generalize from a set of examples to new "unseen" cases. In addition to its remarkable learning powers, the SPA has another significant advantage over ANNs – it can easily representing the knowledge that it has acquired during the learning process in an explicit form. The decision tree format is already a useful form of knowledge representation. Moreover, the SPA can automatically convert its decision trees into sets of production rules. These production rules can be meaningful to the human observer, if there is a meaningful connection between the associated patterns.

Next, we are going to demonstrate how the advantages of the SPA over ANNs in an experimental comparison on the task of learning the past tenses of English verbs.

4 A Performance Comparison of the SPA with ANNs on Learning the Past Tenses of English Verbs

Our main aim in this paper was to answer the challenge posed to the symbolic processing approach by MacWhinney and Leinbach's as well as Rumelhart and McClelland's connectionist models of learning the past tenses of English verbs. So we structured our experiment[5] in such a way as to guarantee a maximal common basis for comparison between their models and the symbolic learning model. Since MacWhinney and Leinbach's used a much larger set of verbs than Rumelhart and McClelland, we borrowed MacWhinney and Leinbach's list of verbs. It contained exactly 1,404 English verb stem/past tense pairs. Among them 1,262 are regular verbs and 142 are irregular ones[6]. Because of the large overlap in the verb sets used in our experiment and by MacWhinney and Leinbach, our results can directly be compared with their results. Even though Rumelhart and McClelland used altogether only 506 verbs, the overlap between the sets is still sufficient for a similar comparison.

[5] Our SPA programs and training and testing data sets can be found in the anonymous ftp site ftp.csd.uwo.ca in the directory pub/spa.

[6] Thus the default accuracy (always predicting past tenses of unseen verbs as regular-suffix adding changes) is $1262/1404 = 89.9\%$. But predicting unseen verbs as regular verbs all the time is not psychologically realistic since human subjects do produce irregular past tenses for unseen verbs. See Section 4.4.2.

4.1 Data Representation Format

We did not use a template/feature representational format and decided to adopt the much simpler left-to-right phonetic coding in the UNIBET format. (We also carried an additional test using the template format to see if that would change significantly our results. See Section 6 for more details.) We did not attempt to use Wickelphone/Wickelfeature format since it was shown to be psychologically and linguistically unjustified by Pinker and Prince's and Lachter and Bever's criticism. Also, unlike MacWhinney and Leinbach's model we did not include any special purpose copying devices that would allow the stem of a verb to be copied directly to the output of our system, i.e. the SPA had to learn that the past tenses of English verbs are a modification of the stem, rather than have this assumption hardwired in.

As it happened, the verbs in MacWhinney and Leinbach list were not longer than 10 phonemes, so we provided 15 input and output attributes, i.e. the SPA in this particular experimental set-up could not handle words longer than 15 phonemes, although, if any such words were found the size of the input and output patterns could easily be increased. The values for each attribute ranged over all phonemes in the UNIBET format. See Table 3 for a few examples of the training patterns ($ means blank space).

Present Tense Form Input Phonetic Patterns	Past Tense Form Output Phonetic Patterns	
I1 I2 I3 I4 I5 .... I15	O1 O2 O3 O4 O5 .... O15	
b E t $ $ .... $	b E t $ $ .... $	(bet-bet)
p 6 r s i v $ ...... $	p 6 r s i v d $ ... $	(perceive-perceived)
r 3 d $ $ .... $	r o d $ $ .... $	(ride-rode)
I n t E n d $ ...... $	I n t E n d I d ... $	(intend-intended)
p r E s $ $ .... $	p r E s t $ $ ...... $	(press-pressed)
. . .	. . .	

Table 3: An small training set of verb stems/past tenses

4.2 Training of the SPA

What is a psychologically realistic way of training the SPA for the English past tense learning task? Can it be modelled realistically as a pattern association task in a supervised learning environment? The main problem is to decide how to interpret the inputs and outputs of a pattern associator which is trying to learn the mappings between English verb stems and their past tenses since the real learning environment does not present the child with such neat input/output pairs directly. In order to guarantee psychological plausibility for their model

Rumelhart and McClelland make several crucial assumptions:

> Although the child would be hearing present and past tenses of all kinds of verbs throughout development, we assume that he or she is only able to learn past tenses for verbs already mastered fairly well in the present tense. This is because the real learning environment does not, in fact, present the child with present-tense/past-tense pairs. Rather, it presents the child with past-tense words in sentences occurring in real-world context. The child would therefore have to generate the appropriate present tense form internally with the aid of the entire sentence and context, and this, we suppose, requires that the child already know the present tense of the word. (Rumelhart and McClelland, 1987, p. 222). (Note: by present tense verbs R&M actually mean verb stems).

So, the psychological plausibility of the modelling of the English past tense learning task with the help of pattern associators hinges on the acceptance of the following assumptions:

- Parental verb stems = child's verb stems. The inputs of verb stems to the pattern associator represent accurately the verb stems in the parental speech environment. The set of parental verb stems is identical with the set of verb stems already known by the child.

- Parental past tenses $\neq$ child's past tenses. The "teacher"-provided correct outputs of verb past tenses to the pattern associator represent accurately the verb past tenses used in the parental speech environment. The set of parental past tenses is not identical with the set of past tenses already known by the child.

Even though these assumptions might be questionable, we will take them for granted for the purposes of our experimental comparison. But we wanted to avoid Rumelhart and McClelland's error in providing only verb types to their model. This is very important because the number of actually uttered verb types in parental speech is different from the number of verb tokens. We also tried to avoid yet another of the shortcomings of Rumelhart and McClelland's model in allowing sharp discontinuities in the supply of regular and irregular verbs to our model. That is why we adopt three additional realistic assumptions:

- Verb tokens are used instead of verb types.

- There are no sharp discontinuities in the supply of regular and irregular verb tokens in parental speech (although, there might be some disparity in the actual verb types.)

- The verb tokens are sampled randomly with replacement according to the Francis-Kucera frequency estimates for English verbs.

In accordance with these additional assumptions we wanted our model to increase gradually the number of acquired verbs (modelling the realistic rate of the child's vocabulary increase) and at the same time to simulate the "average" parental speech environment. (Of course, individual parental speech varies significantly but an estimate of the "average" parent may be given based on Francis-Kucera frequency estimates for English verbs.) In other words, we have to provide our model with a "parental speech environment" in which higher frequency verbs have a higher chance of occurring than lower frequency verbs. At the same time we have to allow the model to gradually increase the number of retained verb stems without sharp discontinuities as in Rumelhart and McClelland's model. In order to achieve this aim, we repeatedly sampled verb tokens from our source set. The chance for a verb to be sampled is directly proportional to its Francis-Kucera frequency estimate. We trained our SPA by providing it initially with 20 verb tokens drawn from the source set and then incrementally increasing the training set with a new 20-verb sample in every training session. At each training step (or "epoch") the SPA updated its decision trees and the number of correctly learned past tenses was recorded for both regular and irregular verbs. The training was stopped when the training set contained 8,000 tokens. At the end of training there were 360 (predominantly low frequency) verb types which have not been sampled during training and which formed the basis for our testing set.

One objection to our training procedure might be that children do not sample verbs randomly, but since each choice added verbs to the training set with probability directly proportional to the Francis-Kucera frequency of the verb in the source set, we feel that it is sufficiently realistic.

4.3 U-shaped Learning Curve

Psychologists generally agree that during the early stages of development of children who grow up in a native English speakers' environment exhibit a special U-shaped learning curve in mastering the English past tense. This phenomenon can be described in the following way: after an early period of correctly producing the past tenses of both regular and irregular verbs, children begin to overregularize the irregular verbs (e.g. *come* $\mapsto$ *comed*). The period of overregularization can last from the age of two well into the school-age years. (Cf. Marcus et al. (1993) for a state of the art discussion.) Previously, it has been argued that the phenomenon of overregularization lends direct support to the symbolic processing approach to cognition in that cognitive processes must be explained in terms of rules and symbolic representations rather than in terms of rote and reinforcement (Slobin, 1971). However, Rumelhart and McClelland claimed that they can model this U-shaped developmental curve with their connectionist pattern associator without the need to postulate the existence of rules and symbolic representations. MacWhinney and Leinbach were not able to model the full U-shaped phenomenon; they reported only a gradual increase in performance accuracy (on irregular verbs only) during training. (Cf. M&L, p. 146).

We were also able to model this U-shaped development by following what we believe to be a more realistic training procedure and by making a much more psychologically realistic assumption about the *causes* of the U-shaped effect than the connectionist models do. As we noted earlier we gradually increased the number of training examples available to the SPA, and we used multiple verb tokens rather than verb types. The training set was grown cumulatively by sampling verb tokens with probability directly proportional to the Francis-Kucera frequency of the verb in the source set. However, at no point there were any sharp discontinuities in vocabulary size, or in the ratio between regular and irregular verb tokens. Initially, the percentage of regular tokens among all tokens was 45% and later it gradually converged toward 60%.

Our modelling of the U-shaped learning effect is based to a large degree on the Blocking and Retrieval Failure hypothesis about the mechanism of overregularization advanced by Marcus et al. (1993). This hypothesis is in turn based on the Blocking hypothesis (Pinker, 1984) which we discuss in detail in Section 5. In short, the Blocking hypothesis postulates the existence of a special blocking mechanism which is turned on when an irregular verb, previously stored in memory, is encountered resulting in the blocking of the rules responsible for the production of suffixed regular pasts (e.g. stem + ed) and in the successful production of the irregular past tense form. The Blocking and Retrieval Failure hypothesis conjectures that the retrieval of an irregular blocks the default rule but children's memory traces are not strong enough to guarantee always a perfect retrieval. Hence, blocking sometimes fails, resulting in overregularized past tense form.

We simulate a similar blocking/deblocking interaction by varying a special *noise allowing parameter* of the SPA/C4.5. The SPA has the capability to allow for m number of noisy data in its decision trees. A small m results in the creation of complex trees that try to account for classes with only a few instances (typically irregular verbs); while a large m treats the class with a small number of instances as noise, so the branches are not built and the future instances of that class are assigned by default to the majority class. For example, even if the SPA encounters only one token of the pair go – went, if m=1, it will create branches that will accommodate this single exceptional pair. If m=2 and again only one token of go – went is encountered by the SPA in the training set, it will not build exception handling branches for this single exceptional token and will classify it together with the majority, i.e. the SPA will associate go – goed. If two tokens of go – went are encountered with m=2, then again the trees to handle that exception will be built. So, generally while a small m has the effect of blocking – irregular verbs are treated as exceptions, the relaxation of the noise-tolerance parameter (larger m) has the effect of deblocking – more branches with small number of exceptional entries are allowed to be disregarded and to be classified as the majority class, i.e. irregular verbs are overregularized.

Presumably, children initially learn most verbs as exceptions, so in the initial stages of our simulation we set m to be very small, thus even with few examples the SPA builds very large trees accounting for all individual verbs. When more regular and irregular verbs are seen children start to generalize. This can be modelled by allowing for a larger m. In accordance with our expectations, when we relaxed m from 2 to 6 in the second phase of the training process during which the available tokens grew from 200 to 300, the SPA began to overregularize some of the irregular verbs that it had learned correctly before. As expected the percentage of correctly learned irregular verbs drops due to overregularizations while the percentage of correctly learned regular verbs stays in the same range. We kept the noise-allowing criterion fixed for the rest of the training process. With the addition of more and more verb tokens, the SPA gradually learned both irregular and regular verbs with 99.2% accuracy. (See Figure 2.) Thus, the effect of U-shaped learning is very clear: The SPA initially learns a relatively small number of regular and irregular verbs correctly; then with the relaxation of its "memory for exceptional patterns" irregular verbs are regularized; later, the branches for exceptional patterns are gradually expanded as more and more irregular tokens are presented. We can model various "degrees" of overgeneralization by increasing m to different numbers. For example, if m is increased only from 2 to 3, we observe less than 10% drop in the accuracy of irregular verb token due to overregularization. This low rate of overregularization is in accordance with the latest experimental results. (Cf. Marcus et al, 1993).

We can also explain very elegantly why less frequent irregular verbs are more likely to be overregularized than more frequent irregular verbs. Clearly the number of tokens of the less frequent irregular verbs are more likely to be smaller than m. if the number of tokens of any irregular verb is less than m, it will be regularized in SPA.

4.4 Testing the Generalization Power of SPA

In terms of overall accuracy during *training* the SPA achieved 99.2% correct rate comparable with those of the connectionist simulations (close to 100% correct) on the training set. However, the *testing* results are crucial in determining whether a learning model can project the acquired knowledge to new unseen cases as human learners are evidently capable of doing. A learning model that fails to exhibit a similar strong tendency to generalize beyond the training data is clearly psychologically unrealistic.

4.4.1 Testing on "Novel" English Verbs

In order to achieve a direct comparison with the connectionist models we carried out a series of tests with our SPA under the same test conditions as required by the testing paradigm employed by Rumelhart and McClelland, and MacWhinney and Leinbach.

In our first test, the SPA was trained on a total of 8,000 verb tokens (4,695 regular and 3,305 irregular, representing 1,038 verb types) sampled according to Francis-Kucera frequency during the incremental training session. The SPA's configuration at the end of the last session

Figure 2: U shape of learning regular/irregular verbs.

was used to predict the past tenses of the verbs in the *test* set. The test set consists of 360 low frequency verbs (21 irregular and 339 regular) that were not previously selected from the source set for training. The SPA correctly predicted the past tenses of 86.4% of all test verbs; as expected its predictive success on the regulars was much better than on the irregulars – 89.4% versus 38.1%, respectively. We carried out the same training/testing procedure several times and the outcome was very similar. These results show that our SPA was able to generalize much better to unseen low-frequency verbs than the connectionist models. Our comparative performance results are summarised on Table 4.

From the table it is clear that the SPA outperformed Rumelhart and McClelland's model on both regular and irregular verbs. MacWhinney and Leinbach's did not report any results on the regular verbs and the overall predictive accuracy of their model cannot be estimated. However, our SPA outperforms L&M's model by a wide margin in a series of head-to-head comparisons reported in the Postscript Section (Section 6).

When we analyze the incorrect prediction of the SPA, we found that 36 out of 339 regular verbs were predicted incorrectly by the SPA. Among these 36, 18 are no-change responses, 15 are vowel-change responses, and only 3 are psychologically unrealistic responses. There are no unrealistic responses to the irregular verbs (see below). This represents only 0.83% of

	R&M	M&L	SPA
Verb types in training	420	1,650	1,038
Verb types in testing	86	13 irregular	360
Percent Correct On Training			
Total	97.0	99.3	99.2
Regulars	98.0	100.0	99.6
Irregulars	95.0	90.7	96.6
Percent Correct On Testing			
Total	56.9 - 73.3	N/A*	86.4
Regulars	66.7 - 83.3	N/A*	89.4
Irregulars	7.1 - 21.4	30.8	38.1

Table 4: A comparison between SPA and the connectionist models
*: Following our criticism MacWhinney reports a new connectionist simulation in which he tested on unseen regular verbs (personal communication). Also see Section 6.

psychologically unrealistic responses on all testing verbs. But what is the reason for the very high error rate of the SPA on the unseen *irregular verbs*? In table 5 we list all of the errors made by the SPA on the irregular test verbs. There the majority of the errors are regularization errors – 84.6% of all errors; there is only one no-change error, and one vowel-change error. There are no strange or nonsensical errors.

MacWhinney and Leinbach did not report the verbs in their testing set, so we cannot make a *direct* comparison with their results. In contrast Rumelhart and McClelland listed all of the errors on the 14 irregular test verbs made by the network. In order to compare directly the performance of our pattern associator with theirs, we retrained our SPA *without* including the 14 irregular verbs used by Rumelhart and McClelland. Then we tested the SPA in predicting their "correct" past tenses. Our results are summarised in Table 6.

As can be seen from the table our SPA had 6 correct responses versus 1 minimum to 3 maximum correct responses of Rumelhart and McClelland's model on the same set. When we add to this the SPA's much better predictive accuracy on regular verbs we can see that the SPA performed much better than R&M 's model under their own testing criteria. The question, however is, are these testing criteria, especially for irregular verbs, psychologically realistic? How can we ask the model to predict something that is by definition unpredictable – irregular verbs?

4.4.2 Testing on Novel English-Like Pseudo-Verbs

One of the main objections against the testing paradigm employed by Rumelhart and McClelland, and MacWhinney and Leinbach is that it is not psychologically realistic. The trouble lies

stem	past tense	SPA's output	Status
bet	bet	bet	Correct
bid	bid	bid	Correct
quit	quit	quit	Correct
kneel	knelt	knelt	Correct
shed	shed	shed	Correct
sling	slung	slung	Correct
sting	stung	stung	Correct
wet	wet	wet	Correct
befall	befell	befalled	Regularized
burst	burst	bursted	Regularized
freeze	froze	freezed	Regularized
mislead	misled	misleaded	Regularized
overhear	overheard	overheared	Regularized
overtake	overtook	overtaked	Regularized
split	split	splitted	Regularized
string	strung	strang	Vowel-Change
undergo	underwent	undergoed	Regularized
undo	undid	undoed	Regularized
wind	wound	wind	No-Change
withhold	withheld	withhelded	Regularized
withsatand	withstood	withstanded	Regularized

Table 5: Responses of SPA on unseen irregular verbs

in the definition of the term *novel verb*. If we consider the low frequency regular verbs as truly novel verbs for the model, how are we to determine what should count as a "correct" past tense production. It seems easy to accept any suffixed form of a regular low frequency verb (stem + ed) as a correct output of the model. This would be evidence that the model has acquired an important regularity in the production of the English past tense. However, such reasoning cannot be extended to low frequency irregular verbs without serious qualifications, since by definition irregular verbs are supposed to be unpredictable. For example, if you were asked to produce the past tense form of the English verb *to shend* (meaning to shame), you are most likely to say *shended*. You will be wrong since the correct form is *shent*. But if, as is most likely, you have never heard this verb before, is there a way you could predict the correct past tense form? (Cf. Bybee and Slobin, 1982b, Marcus et al., 1993).

If the answer to this question is uncertain, how are we to judge the reaction of a learning model which chooses to treat the past tense of *cling* as regular *clinged*, or as a no change irregular *cling*, or as an irregular with a different vowel change *clang*? Is there a *truly objective* way to say that any of these choices made by the learning model are correct or incorrect since

R&M's Model	C/I	SPA	C/I
bid $\mapsto$ bid	C	bid $\mapsto$ bid	C
thrust $\mapsto$ thrusted	I	thrust $\mapsto$ thrusted	I
bend $\mapsto$ bended	I	bend $\mapsto$ bend	I
lend $\mapsto$ lended	I	lend $\mapsto$ lend	I
creep $\mapsto$ creeped	I	creep $\mapsto$ crept	C
weep $\mapsto$ weeped	I	weep $\mapsto$ wept	C
weep $\mapsto$ wept	C		
catch $\mapsto$ catched	I	catch $\mapsto$ catched	I
breed $\mapsto$ breeded	I	breed $\mapsto$ bred	C
grind $\mapsto$ grind	I	grind $\mapsto$ grinded	I
wind $\mapsto$ wind	I	wind $\mapsto$ winded	I
cling $\mapsto$ clinged	I	cling $\mapsto$ clung	C
cling $\mapsto$ clung	C		
dig $\mapsto$ digged	I	dig $\mapsto$ digged	I
stick $\mapsto$ sticked	I	stick $\mapsto$ sticked	I
tear $\mapsto$ teared	I	tear $\mapsto$ tore	C

Table 6: Comparison of testing 14 irregular verbs

we are not sure how a native speaker for whom *cling* is a completely novel verb would react. If we want to test the true generalization capabilities of Rumelhart and McClelland's model, we have to somehow get rid of the "benefit of exception memorization" that native speakers enjoy with respect to known irregular verbs and test the predictions of the model against their "uncontaminated" intuitions in producing the past tenses of genuinely novel verbs. One possible solution to this "genuine novel verbs problem" has been offered by Prasada and Pinker who designed a special pseudo-verb testing paradigm (Prasada and Pinker, 1993).

Prasada and Pinker created 60 English-like *pseudo*-verbs. The verbs are divided into two main groups – irregular-like and regular-like pseudo-verbs. Each of these groups are subdivided into three classes – *Prototypical*, *Intermediate*, and *Distant*. Items in the *Prototypical pseudo-irregular* class are similar to several clusters of English irregular verbs involving four typical vowel transformations (in UNIBET form) – [IN $\mapsto$ 6N], [i $\mapsto$ E], [o $\mapsto$ u], and [er $\mapsto$ or]. The *Intermediate pseudo-irregular* class is derived by taking each typical pseudo-irregular item and changing either the initial or the final consonant cluster. The *Distant pseudo-irregulars* are produced in the same way, but by changing *both* the initial and the final consonant clusters. The items in the *Prototypical pseudo-regular* class are designed to be similar to several high frequency regular verbs with one condition – they include the same vowels used in the pseudo-irregular items. The Intermediate and Distant pseudo-regulars are produced in the same way as the Intermediate and Distant pseudo-irregulars.

This pseudo-verb testing paradigm allows for a direct and psychologically relevant comparison between the "pure" intuitions of native speakers about English past tenses and the predictions of a computational learning model on verbs that are *truly novel* for both the humans and the model. Prasada and Pinker carried out a test of Rumelhart and McClelland's model using this pseudo-verb paradigm. They trained the network on the 422 verbs (341 regular and 81 irregular) which were used in Rumelhart and McClelland's simulation. The model was trained for 200 epochs and then was tested on the set of pseudo-verbs. Because the simulation occasionally output bizarre strings or normal English verbs (not pseudo verbs!) Prasada and Pinker decided to interpret liberally the network's responses. They counted any no-change or nonsuffixed vowel-changing response as correct pseudo-past tense production. From the pseudo-irregular items the network produced: 3 pseudo-irregular past forms from the *Prototypical* items; 1 past from the *Intermediate* items; and none from the *Distant* items. For the pseudo-regular verbs, Prasada and Pinker counted any suffixed pseudo-past tense productions as correct. The network produced 7, 4, and 1 suffixed pseudo-past forms for the *Prototypical*, *Intermediate*, and *Distant* sets, respectively.

In order to assess the psychological validity of these results, Prasada and Pinker carried out an experiment with 24 native English speakers. They wanted to test specifically the likelihood that native speakers' novel pseudo-verbs would be similar to the decisions of Rumelhart and McClelland's model in producing regular and irregular past tense forms of the pseudo-stems, so that it would be possible to compare directly the human results with the connectionist simulations. The subjects were familiarized with the "meanings" of 6 of the pseudo-verbs at a time and were told to fill the blanks in a text with the appropriate pseudo verbs. The context called for a past tense form. The average number of nonsuffixed irregular past responses (vowel change or no change responses) were 5.9, 4.6, and 3.7 for the Prototypical, Intermediate, and Distant pseudo-irregular classes. The average number of suffixed (-ed) regular past responses were 9.5, 9.2, and 8.9 for the Prototypical, Intermediate, and Distant pseudo-regular classes. As we can see (also on FIGURE 3 (a) and (b))the difference between Rumelhart and McClelland's simulation and human subjects' responses is quite large.

We wanted to test the inductive generalization capabilities of the SPA in a psychologically relevant experiment, so that our results can be realistically compared with human subjects and with Rumelhart and McClelland's connectionist model. We performed the following experiment: We trained the SPA on all 1,404 of the verbs in the original verb set. Then we tested the SPA on the 60 pseudo-verbs used in Prasada and Pinker's experiment. All of the past tense forms produced by the SPA were "good" pseudo-past forms, i.e. in contrast to the Rumelhart and McClelland's model, no normal English verbs and no bizarre consonant changes or strange suffixes were produced. The actual past tense forms produced by the SPA are shown in Table 7.

Irregular Pseudo-Verbs				
Pseudo-Verb Stem		SPA Output		Status
Spelling Form	UNIBET Form	Spelling Form	UNIBET Form	
Prototypical Pseudo-Irregular: 5 Pseudo-Irregular				
spling	splIN	splang	spl&N	I
skring	skrIN	skrung	skr6N	I
sprink	sprINk	sprink	sprINk	I
cleed	klid	cleeded	klidId	R
preed	prid	pred	prEd	I
queed	kwid	queeded	kwidId	R
cloe	klo	cloed	klod	R
froe	fro	froed	frod	R
plare	plEr	plared	plErd	R
quare	kwEr	quore	kwor	I
Intermediate Pseudo-Irregular: 3 Pseudo-Irregulars				
fring	frIN	frung	fr6N	I
ning	nIN	ning	nIN	I
frink	frINk	frunk	fr6Nk	I
cleef	klif	cleefed	klift	R
preek	prik	preeked	prikt	R
queef	kwif	queefed	kwift	R
foa	fo	foaed	fod	R
voa	vo	voaed	vod	R
jare	dZEr	jared	dZErd	R
grare	grEr	grared	grErd	R
Distant Pseudo-Irregular: 3 Pseudo-Irregular				
trisp	trIsp	trispped	trIspt	R
nist	nIst	nisted	nIstId	R
blip	blIp	blipped	blIpt	R
gleef	glif	gleefed	glift	R
keeb	kib	kebt	kEbd	I
meep	mip	mept	mEpt	I
goav	gov	wev	wEv	I*
joam	dZom	joamed	dZomd	R
flape	flep	flaped	flept	R
blafe	blef	blafed	bleft	R

Regular Pseudo-Verbs				
Pseudo-Verb Stem		SPA Output		Status
Spelling Form	UNIBET Form	Spelling Form	UNIBET Form	
Prototypical Pseudo-Regular: 10 Pseudo-Regulars				
plip	plIp	plipped	plIpt	R
glip	glIp	glipped	glIpt	R
brip	brIp	bripped	brIpt	R
gloke	glok	gloked	glokt	R
proke	prok	proked	prokt	R
greem	grim	greemed	grimd	R
pleem	plim	plimed	plimd	R
treem	trim	treemed	trimd	R
slace	sles	slaced	slest	R
nace	nes	naced	nest	R
Intermediate Pseudo-Regular: 7 Pseudo-Regulars				
brilth	brIlD	brilthed	brIlDd	R
glinth	glInD	glinthed	glInDd	R
plimph	plImf	plimphed	plImft	R
ploab	plob	ploabed	plobd	R
ploag	plog	ploaged	plogd	R
smeeb	smib	smebt	smEbd	I
smeeg	smig	smegt	smEgd	I
smeej	smidZ	smejt	smEdZd	I
smaib	smeb	smaibed	smebd	R
smaig	smeg	smaiged	smegd	R
Distant Pseudo-Regular: 7 Pseudo-Regulars				
frilg	frIlg	frilged	frIlgd	R
krilg	krIlg	krilged	krIlgd	R
trilb	trIlb	trilbed	trIlbd	R
ploamph	plomf	ploamphed	plomft	R
ploanth	plonD	ploanthed	plonDd	R
smeelth	smilD	smeltht	smElDd	I
smeenth	sminD	smentht	smEnDd	I
smeerg	smirg	smergt	smErgd	I
smairg	smerg	smairged	smergd	R
smairph	smerf	smairphed	smerft	R

* Interesting answer: due to the go $\mapsto$ went training pair.

Table 7: SPA's responses on Pseudo-verbs from Prasada and Pinker (1992)

What is remarkable about these results is that the SPA came much closer to the actual native speakers' responses than the connectionist pattern associator. See Figure 3. In particular, the SPA shadowed very closely the native speakers' responses to *Pseudo-Irregular* verbs in that it treated more of the *Prototypical*, *Intermediate*, and *Distant* pseudo-irregulars as irregulars than the connectionist pattern associator. But the difference in the performance of the SPA and Rumelhart and McClelland's model is much more obvious with respect to the set of *pseudo-regular* verbs. There the SPA again was very close to the actual human responses but the predictions of the connectionist pattern associator deviate significantly from the human results. These comparative results indicate that the SPA can exploit the similarities between irregular sounding phonetic patterns slightly better than the connectionist model but that it is undeniably much more powerful when it comes to generalizing to the regular pasts.

Figure 3: (a) Pseudo-Irregular Verbs (b) Pseudo-Regular Verbs

4.5 Learning of the Full English Verb Paradigm

We decided to verify whether the SPA can be used successfully to learn the full English verb paradigm. Initially, we ran three separate tests on learning third person singular, present participle, and past participle. And finally we carried out another test with all verb forms – third person singular, past tense, present participle, and past participle. In this test only, category information for third person singular, past tense, present participle, and past participle is directly encoded alongside with phonetic information. Our results are summarized in Table 8. Note that the training and testing sets are completely disjoint.

Verb Form	Verb Set Size		Accuracy	
	Training	Testing	Training	Testing
3rd person singular	500	100	100%	96.0%
present participle	800	500	100%	97.8%
past participle	1,000	300	100%	84.3%
all + past tense	3,000	1,000	100%	85.9%

Table 8: Learning of the full English verb Paradigm

These results indicate that the SPA is indeed able to learn to associate verb stems to their present tenses (3d person singular), present participles, and past participles, in addition to their past tenses. The first two results speak for themselves. Since the forms for 3d person singular and present participle contain almost no irregularities, the SPA is able to learn to predict them with almost 100% accuracy even from the relatively small training sets. This demonstrates that the SPA has learned the relevant rules necessary to project from a limited training set to all English verbs. The results for past participle are lower because all irregular forms that could not be predicted by the SPA were counted as errors.

5 Explicit Representation and Further Generalization

We consider the comparative results presented so far to be a convincing proof that the SPA outperforms both MacWhinney and Leinbach's, and Rumelhart and McClelland's connectionist models in the task of learning the past tenses of the English verbs. See Section 6 an ultimate comparison of the raw generalization power between the SPA and ANN simulation on this task. However, symbolic learning systems such as our SPA have an additional significant advantage over the connectionist learning models. In contrast to the complete absence of explicit knowledge representation capacities in connectionist pattern associators, the SPA can express the acquired knowledge in the form of *production rules*. These rules are usually sensible and understandable to humans, and can generate explanations of what is learned and how a particular response is produced. Such explicit representation and explanations increase the credibility of the system in learning complicated tasks. There is another great advantage of explicit knowledge representation in learning verb past tenses: It opens the possibility for higher level processing and integration of these rules to lexical, morphological, and syntactic levels. Thus, unlike the connectionist models, the SPA does not leave the integration of the acquired linguistic knowledge as a complete mystery.

5.1 Tree Structure and Ordered Rules

If we look at the internal structure of the decision trees formed during the training process, we can observe that the SPA discovered that usually English verbs change very little from present to past tense, i.e. most of the branches of the SPA's trees were simple identity-preserving branches; they just passed the same value of the examined attribute to the terminal leaf. All trees developed a similar overall structure with many isomorphic branches and subbranches. Figure 4 shows a typical decision tree which determines the values of one of the attributes in the output patterns.

Figure 4: Three functionally distinct substructures

All branches (on all trees) developed by the SPA as the result of the training process could broadly be grouped into three main functionally discrete clusters – *exception-handling* branches, suffix-adding branches, and a large group of *identity-preserving* branches. Identity preserving branches simply pass the value of the encountered phoneme as the value of the corresponding output attribute in the output pattern. The suffix-adding branches, are slightly more complex; they are concerned with examining these phoneme positions where a phoneme was followed with a blank $ sign, i.e. verb endings. These branches result in the assignment of the regular past suffixes /d/, /t/ or /Id/ to the output pattern. But if certain "exceptional sounding" phonemes are encountered at pre-specified positions in the input pattern, the trees branch out in different direction – suffixation is not used but an "exceptional" phonetic pattern is produced, i.e. the SPA is learning/predicting an irregular verb. Note, that the exception-handling branches generally have more intermediate nodes than suffix-adding branches, and both suffix-adding branches and exception-handling branches have more intermediate nodes than the identity-preserving branches which have just one. (Note, that these functionally distinct subgroups have formed as a result of the training process; if the SPA was trained on some other data, different structure might result.)

As we saw already, a decision tree can be converted directly and without any loss of generality into a set of production rules. This is done algorithmically by tracing each set of branches starting from a terminal leaf through all intermediate nodes up to the root of the tree. This procedure is implemented in the SPA without making any distinction between different branch lengths. However, the fact that the SPA formed three functionally distinct groups of branches can be usefully exploited during the conversion process from the decision tree format, into the rule format. In a slightly different implementation we can make the SPA convert into rules by tracing *first* the longest tree branches which have the greatest number of antecedent nodes which usually classify components of irregular verbs; then the next longest branch, and so on until all the branches are exhausted. This conversion-simplification procedure will result in a set of *precedence-ordered* rules, namely exception-handling rules, suffix-adding rules, and identity rules. The rules are ordered in the sense that only the *first* rule in the ordered list whose conditions are satisfied would fire.

After the conversion and simplification of the trees of the SPA is complete, we are left with a precedence-ordered set of propositional phonetic rules. (See Table 9.) The exception-handling rules listed first in the precedence-ordering are clearly *propositional* phonetic rules. By propositional phonetic rules we mean rules that apply only over positioned phonemes. For example, the following is a typical suffix-adding rule in propositional phonetic form:

If $\mathbf{I}_4$ = z and $\mathbf{I}_5$ = \$, then $\mathbf{O}_5$ = d

It expresses the following information: Look at the fourth and at the fifth phonemes of an input verb stem, if the fourth phoneme is /z/ and the fifth is blank /\$/ (i.e. if we are at a verb ending) then add /d/ as the fifth phoneme to the output form of the verb. Generally, the suffix-adding rules have only two antecedents, while the identity rules have only one. A typical identity rule looks like this:

If $\mathbf{I}_5$ = z, then $\mathbf{O}_5$ = z

An important characteristic of this ordering is that the rules with a higher precedence are also the most specific and the rules with a lower precedence are more general. Immediately after the first group of exceptional propositional rules for irregular verbs come a set of rules which have a common structure – they are all rules that have a body consisting of one phoneme followed by a blank /\$/, (i.e. if this is the ending of a verb) and a consequent consisting of the regular suffix -ed (/d/, /t/ or /Id/). They are followed by the most unspecific rules – the identity-mapping rules.

Ordered Rule types	Rules for fourth output	Rules for fifth output
Exception-Handling rules (See Section 5.2)	If $\mathbf{I}_2$ = p & $\mathbf{I}_4$ = I & $\mathbf{I}_5$ = N & $\mathbf{I}_6$ = \$ then $\mathbf{O}_4$ = & (spring $\mapsto$ sprang)	If $\mathbf{I}_4$ = N & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = \$ (ending for swing $\mapsto$ swung) (See Table 10) ......
	If $\mathbf{I}_3$ = t & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = I	If $\mathbf{I}_4$ = t & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = I
	If $\mathbf{I}_2$ = t & $\mathbf{I}_3$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_3$ = t & $\mathbf{I}_4$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = d & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = I	If $\mathbf{I}_4$ = d & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = I
	If $\mathbf{I}_2$ = d & $\mathbf{I}_3$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_3$ = d & $\mathbf{I}_4$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = k & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = t	If $\mathbf{I}_4$ = k & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = t
	If $\mathbf{I}_3$ = p & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = t	If $\mathbf{I}_4$ = p & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = t
	If $\mathbf{I}_3$ = s & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = t	If $\mathbf{I}_4$ = s & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = t
	If $\mathbf{I}_3$ = S & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = t	If $\mathbf{I}_4$ = S & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = t
	If $\mathbf{I}_3$ = f & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = t	If $\mathbf{I}_4$ = f & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = t
	If $\mathbf{I}_3$ = r & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = r & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = m & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = m & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
Suffix-adding rules (See Section 5.3)	If $\mathbf{I}_3$ = n & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = n & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = l & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = l & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = N & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = N & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = g & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = g & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = z & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = z & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = b & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = b & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = D & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = D & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = v & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = v & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = u & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = u & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = e & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = e & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = i & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = i & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = o & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = o & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = 3 & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = 3 & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_3$ = 1 & $\mathbf{I}_4$ = \$ then $\mathbf{O}_4$ = d	If $\mathbf{I}_4$ = 1 & $\mathbf{I}_5$ = \$ then $\mathbf{O}_5$ = d
	If $\mathbf{I}_4$ = p then $\mathbf{O}_4$ = p	If $\mathbf{I}_5$ = p then $\mathbf{O}_5$ = p
	If $\mathbf{I}_4$ = r then $\mathbf{O}_4$ = r	If $\mathbf{I}_5$ = r then $\mathbf{O}_5$ = r
Identity-Mapping Rules (See Section 5.3)	If $\mathbf{I}_4$ = u then $\mathbf{O}_4$ = u	If $\mathbf{I}_5$ = u then $\mathbf{O}_5$ = u
	If $\mathbf{I}_4$ = i then $\mathbf{O}_4$ = i	If $\mathbf{I}_5$ = i then $\mathbf{O}_5$ = i
	If $\mathbf{I}_4$ = 3 then $\mathbf{O}_4$ = 3	If $\mathbf{I}_5$ = 3 then $\mathbf{O}_5$ = 3
	......	......

Table 9: Ordered rules produced from decision tree 4 and 5.

It is very interesting to observe that this precedence-ordering of the rules produced by the SPA has the same effect as the Blocking Principle – an exception-handling rule would fire first if an irregular verb is encountered, thus blocking the possibility of the suffix-adding rules to fire. We discuss the implications of this effect in the following sections.

5.2 Associative Templates for Irregular Verbs

If the training patterns are not self contradictory, it is always possible to construct decision trees/rules that reproduce *all* output patterns in the training set correctly. However, the SPA constructs small decision trees and rule sets that capture *minimally discriminative* or *most general* information needed for classifying a given set of patters. In the application of learning the past tenses of English verbs, these rules are just *specific enough* to classify a *given* set of regular and irregular verbs. Thus, they are in some sense *necessary* and *sufficient* conditions for distinguishing and classifying a given set of verbs. Since exception-handling rules are minimally discriminative for irregular verbs, they are more general than rote learning in which each irregular verb is individually memorized and prediction of other irregular verbs is not possible.

Let us look at how exactly the SPA learns most general rules to associate the irregular pair *swing* $\mapsto$ *swung*. (At this point, *bring* $\mapsto$ *brought* is not in the training set.) If we trace the rules that have fired to produce this transformation we receive the set of rules in Table 10.

```
SPA's        SPA's
Input        Output
s w I N $ -> s w 6 N $  (UNIBET coding for swing -> swung)
1 2 3 4 5    1 2 3 4 5  (attribute positions)
s * * * * -> s          (identity-mapping rule: If i1 = s, then o1 = s)
* w * * * ->   w        (identity-mapping rule: If i2 = w, then o2 = w)
* * I N $ ->     6      (exception: If i3 = I & i4 = N & i5 = $, then o3 = 6)
* * * N * ->       N    (identity-mapping rule: If i4 = N, then o4 = N)
* * * N $ ->         $  (exception: If i4 = N & i5 = $, then o5 = $)
```

Table 10: Rules for *swing* $\mapsto$ *swung*.

And if we further trace the rules for *cling* $\mapsto$ *clung*, *sling* $\mapsto$ *slung*, and *fling* $\mapsto$ *flung*, we receive the following response: (See Table 11.)

We can clearly see that in the four cases above the irregular cluster "**IN$" was encountered which led to the firing of the two exceptional rules: an exceptional phoneme-changing rule from tree #3 and a suffix-blocking rule from tree #5). The two exceptional rules for these particular positions are functionally identical to an *associative template* that can account for

cling --->	clung	sling --->	slung	fling --->	flung
SPA's	SPA's	SPA's	SPA's	SPA's	SPA's
Input	Output	Input	Output	Input	Output
k l I N $ ->	k l 6 N $	s l I N $ ->	s l 6 N $	f l I N $ ->	f l 6 N $
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
k * * * * ->	k	s * * * * ->	s	f * * * * ->	f
* l * * * ->	l	* l * * * ->	l	* l * * * ->	l
* * I N $ ->	6	* * I N $ ->	6	* * I N $ ->	6
* * * N * ->	N	* * * N * ->	N	* * * N * ->	N
* * * N $ ->	$	* * * N $ ->	$	* * * N $ ->	$

Table 11: Rules for *cling* ↦ *clung*, *sling* ↦ *slung*, *fling* ↦ *flung*.

several irregular forms. (We use bold capital letters to represent the empty identity preserving consonant/vowel slots of the template where any phoneme could be substituted):

X Y I N \$ ↦ **X Y** 6 N \$.

As we can see, the SPA constructs the most general rules which contain as many phoneme variables as possible for several "similar" irregular verbs from the given training set.

Even if a template constructed during training is for *one* irregular verb only, such a template is usually more general than rote learning. For example, we found that *spring* is classified into *sprang* in the training set by rules forming the following template:

X p **Y** I N \$ ↦ **X** p **Y** & N \$.

Although this template is only applicable to the mapping *spring* ↦ *sprang*, it is still more general than rote learning. It can be used to predict other pseudo-verbs. For example, by using this template our SPA classified the pseudo-verb spling (splIN) into splang (spl&N), which is a very meaningful prediction. For the very same reason, our SPA predicted the past tense of the pseudo-verb *goav (gov)* to be *wev (wEv)*[7]. Similar associative templates caused the irregular productions for *spling* ↦ *splang, skring* ↦ *skrung, meep* ↦ *mept, smeeb* ↦ *smebt, smeeg* ↦ *smegt* etc. Thus, to us the human-like responses of the SPA to Prasada and Pinker's pseudo-verbs is *not a mystery*. We can account for each individual response and find out which irregular associations caused the blocking of the regular suffix-adding rules.

The possible existence of such irregular associative templates (or clusters of weakly generative propositional rules) may explain the exact mechanism which implements the *Blocking Principle*. The associative propositional templates have a clear precedence over suffix-adding rules. Every incoming verb is matched against the templates and if the verb contains an exception triggering sequence like "**IN\$", the application of the lower-precedence suffix-adding

[7]In several talks on this subject we asked members of the audience to predict the past tenses of several pseudo-verbs including *goav*. Every time there were some individuals who responded with *wev*.

rules is blocked. If none of the templates fire, then the suffix-adding rules apply.

Note that these associative templates are extremely flexible and can accommodate changes in the associative connections among irregular verbs. As new exceptional forms are learned by the child or dropped by infrequent use, some templates become more specific or more general. For example, a child who has already acquired the template

$\mathbf{X}\ \mathbf{Y}\ \mathrm{I\ N\ \$} \mapsto \mathbf{X}\ \mathbf{Y}\ \mathrm{6\ N\ \$}$.

and who encounters the verb *to bring* for the first time is likely to produce *brung* as its past tense form the first time. Such errors occur with non-zero probability in young children and some English dialects actually use this past tense form. (Cf. Pinker and Prince, 1991). But when a child learns that the correct transformation is *bring* $\mapsto$ *brought*, he/she has to accommodate this exception among the existing templates.

The SPA can very easily model this process. If a new irregular verb is encountered by the SPA during training, the decision trees will be updated. This results in splitting the old overly general template in two new more specific templates, or if the new verb is completely irregular, like *go* $\mapsto$ *went*, it is encoded individually as a new template. For example, after the SPA has produced the template that associates together *swing* $\mapsto$ *swung*, *cling* $\mapsto$ *clung*, *sling* $\mapsto$ *slung*, and *fling* $\mapsto$ *flung*, it was presented with the pair *bring* $\mapsto$ *brought*. The original template was immediately split in two more specific templates:

$\mathbf{X}\ \mathrm{l\ I\ N\ \$} \mapsto \mathbf{X}\ \mathrm{l\ 6\ N\ \$}$

accounting for *cling* $\mapsto$ *clung*, *sling* $\mapsto$ *slung*, and *fling* $\mapsto$ *flung*; and to

$\mathrm{s}\ \mathbf{X}\ \mathrm{I\ N\ \$} \mapsto \mathrm{s}\ \mathbf{X}\ \mathrm{6\ N\ \$}$

accounting for *swing* $\mapsto$ *swung*, *sting* $\mapsto$ *stung*. Thus the possible conflict with *bring* $\mapsto$ *brought* is avoided.

The fact that the associative templates produced by the SPA are flexible in scope (i.e. can be specified or generalized) is very important if we want to account for historical changes in irregular morphology. Languages are constantly creating/destroying similar sounding patterns and we need to be able to explain how new irregular/regularized forms are affected by the already existing similarities known by the speakers of that language.

Our conjecture about the existence of special associative templates which control the production of irregular past tense forms is similar in spirit to Pinker and Prince's hypothesis that the processing of irregular verbs is done in a special part of human associative memory (Pinker and Prince, 1991; Pinker, 1991). However, Pinker and Prince suppose that this type of memory can be broadly characterized in terms of the associative properties of artificial neural networks. Our proposal is somewhat different in that we suppose that *symbolic* phoneme/position templates can have the properties of human associative memory. In this sense our working model is closer to traditional rule-based accounts of human linguistic competence.

5.3 Rules for Regular Verbs and Further Generalization

We saw that the ordering of the propositional rules produced by the SPA represents a strict precedence order which can be used to explain the Blocking Principle. Since the most specific exception-handling rules have clear precedence over the rest, if none of the most specific rules fire first, the regular suffix-adding rules can fire second. But if they do not fire, the identity-mapping rules can fire last. For example, the following rules fire to produce the past tense of the regular verb *to enclose* (Inkloz). (See Table 12.) In all but the last rule, identity mapping

```
SPA's         SPA's
Input         Output
I n k l o z $  -> I n k l o z d (UNIBET coding for enclose -> enclosed)
1 2 3 4 5 6 7     1 2 3 4 5 6 7    (attribute positions)
I * * * * * *  -> I                (identity-mapping rule from tree #1)
* n * * * * *  ->   n              (identity-mapping rule from tree #2)
* * k * * * *  -      k            (identity-mapping rule from tree #3)
* * * l * * *  ->       l          (identity-mapping rule from tree #4)
* * * * o * *  ->         o        (identity-mapping rule from tree #5)
* * * * * z *  ->           z      (identity-mapping rule from tree #6)
* * * * * z $  ->             d    (suffix-adding rule from tree #7)
```

Table 12: Rules for *enclose* $\mapsto$ *enclosed*.

rules fire because there are no irregular rules or regular suffix-adding rules to block them. The last rule fires because there is no exceptional rules to stop suffix-adding rules from firing.

However, there are some problems with this explanation. As we noted the regular suffix-adding rules are also propositional phonetic rules, like the exceptional ones. But it is psychologically unrealistic to assume that the human linguistic system can support so many structurally similar rules without attempting to generalize them. A very reasonable conjecture is that the processing of regular verbs does not remain forever tied to the level of position-relevant phonetic templates (propositional rule clusters), as is the case with irregular verbs, but that humans generalize the regularities among regular verbs to a much higher (possibly first-order) level. Can the SPA similarly compress and generalize the numerous propositional phonetic rules that govern the addition of regular suffixes? The present implementation of the SPA cannot go beyond the propositional level of representation. However, the relevant information has already been extracted in explicit form from the training samples and can algorithmically be converted into first order format.

We next demonstrate how this further generalization can be done algorithmically in a peripheral extension of the core SPA. We do not claim that exactly the same inductive steps are taken by human learners, however, the rules resulting from the generalizations carried out algorithmically using the following two strategies seem to be psychologically realistic.

The "Horizontal" Compression Strategy Applying the LGG algorithm (*least general generalization*) (Cf. Plotkin, 1970) on the *indexes* of the attributes produces one rule that is the least general generalization of two adjacent regular rules for one and the same final phoneme. For example if we take two adjacent rules for /p/, we have:

If $\mathbf{I}_4$ = p and $\mathbf{I}_5$ = \$, then $\mathbf{O}_5$ = t

If $\mathbf{I}_5$ = p and $\mathbf{I}_6$ = \$, then $\mathbf{O}_6$ = t

The least general generalization of these two rules is the rule:

For all n, If $\mathbf{I}_n$ = p and $\mathbf{I}_{n+1}$ = \$, then $\mathbf{O}_{n+1}$ = t

This is a first order phonetic rule which states that for *all* phoneme positions, if /p/ is a phoneme in the input phonetic pattern, and if it is final (i.e. if it is followed by a blank), the final phoneme in the output phonetic pattern has to be /t/. We generalize all other suffix-adding phonemes in the same way by using the LGG algorithm. Here is a partial list of our results:

For all n, If $\mathbf{I}_n$ = t and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+1}$ = I

For all n, If $\mathbf{I}_n$ = t and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+2}$ = d

......

For all n, If $\mathbf{I}_n$ = p and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+1}$ = t

For all n, If $\mathbf{I}_n$ = k and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+1}$ = t

......

For all n, If $\mathbf{I}_n$ = r and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+1}$ = d

For all n, If $\mathbf{I}_n$ = o and $\mathbf{I}_{n+1}$ = \$ then $\mathbf{O}_{n+1}$ = d

......

The "Vertical" Compression Strategy The new rules that we now have are much more general than the old ones – they are *position irrelevant* phonetic rules. But we are still missing some significant generalizations. Is it realistic to suppose that the human linguistic system represents all regular rules individualized per specific phonemes? Obviously, further "vertical" compression is possible. If we allow the system to create new predicates for disjunctions (Muggleton and Buntine, 1988; Lapointe, Ling, & Matwin, 1993) (i.e. to invent necessary new predicates), we can algorithmically convert all the phonemes that control, for example, the suffix /t/ into one new predicate; let us call it *newpred.* Then we can have the following simple rules for the suffix t:

For all n, If $\mathbf{I}_n$ = newpred and $\mathbf{I}_{n+1}$ = \$, then $\mathbf{O}_{n+1}$ = t

newpred = {p or k or s or S or f}

In this sense, newpred can be considered as extensionally equivalent to the predicate "an unvoiced consonant" In the same way we produce the rules for the other two regular suffixes. The order of generalization procedures is immaterial. The rules can be algorithmically compressed first vertically and then horizontally, or vice-versa.

The same generalization procedures can be applied to the identity rules to produce one very general identity rule:

For all n, If $\mathbf{I}_n = \mathbf{X}$ then $\mathbf{O}_n = \mathbf{X}$

$\mathbf{X}$ is any phoneme.

It is important to note, however, that the above inductive algorithms cannot compress or generalize most of the irregular verb templates. Therefore, the exceptional information encoded in these templates will remain in *propositional* form.

5.4 Modules for the Processing of Regular and Irregular Verbs

If we consider the first-order rules induced by our extended model that form the *default rules for regular past tenses* and the propositional level rules induced by the SPA that form the *associative templates for irregular verbs*, we can clearly distinguish the outlines of two functionally distinct language processing *(sub)modules*. See Table 13.

Ordered Rule types	Rules
Associative templates	**X** l I N \$ $\mapsto$ **X** l 6 N \$ s **X** I N \$ $\mapsto$ s **X** 6 N \$. . . . . . .
Default Rules	For all **X** (strings of any length): If **X** $\circ$ newpred1 $\mapsto$ **X** $\circ$ newpred1 $\circ$ Id newpred1 = {t or d} (For example, *IkstEnd (extend)* $\mapsto$ *IkstEndId (extended)*) For all **X** (strings of any length): If **X** $\circ$ newpred2 $\mapsto$ **X** $\circ$ newpred2 $\circ$ t newpred2 = {k or p or s or S or f } (For example, *tOk (talk)* $\mapsto$ *tOkt (talked)*) For all **X** (strings of any length): If **X** $\circ$ newpred3 $\mapsto$ **X** $\circ$ newpred3 $\circ$ d newpred3 = {b or g or l or m or n or N or v or z or D or e or i or u or o or 1 or 3 } (For example, *salv (solve)* $\mapsto$ *salvd (solved)*) Note: $\circ$ is the concatenation symbol.

Table 13: Two Functionally Distinct Language Processing (Sub)modules.

The two (sub)modules together serve to *distinguish* regular from irregular verbs, and to *predict* the past tenses of unseen verbs, *unmarked* as regular or irregular. Generally speaking, both of these (sub)modules are similar in the sense that both regular and irregular verbs are computed by production rules. After all, irregular verbs can be regarded as regular with "rules" applicable to only a few verbs. However, these rules are highly *heterogeneous* – the clusters of rules that form the associative templates are clearly positioned phoneme rules, and they are very specific for a few or sometimes for one verbs only. On the other hand, the default rules produced by the extended SPA are clearly first-order rules that apply over strings of arbitrary length. They are very simple and general. This is one of the main differences underlying the existence of two functionally distinct (sub)modules responsible for the separate processing of irregular and regular verbs. There is a growing body of psychological and neurophysiological evidence that supports the existence of such functionally distinct (sub)modules for the processing of regular and irregular forms. (Cf. Pinker and Prince, 1991).

The existence of precedence-ordered (sub)modules of propositional-level templates and first-order rules can provide an even more plausible mechanism for implementing the Blocking Principle. (Cf. Kiparsky, 1982; Aronoff, 1976; Pinker and Prince, 1991, Marcus et al., 1993). The successful matching of a verb stem into one of the associative templates has the effect of *blocking* the firing of the default rules. On the other hand, the "loosening" of these associative templates as a result of memory failure, or infrequent usage, may cause a failure to block, resulting in the production of an overregularized form. Thus, our model can give an explanation of the mechanism behind the the famous U-shaped effect that is very much similar to Marcus et al. 1991 explanation of this effect.

Despite our doubts about how well our compression algorithms model the actual psychological process of learning English past tense, the final set of rules including the abstract representation of the verb stem and the verb endings using new predicates in Table 13 seem to be a psychologically realistic.

One problem with our model might be in accounting for the real-time implementation of blocking (Cf. Pinker and Prince, 1991). The problem is that if a precedence ordering is assumed between the two modules, i.e. if during processing a verb stem is first checked against several templates, and if only after this, it is passed on to the default rules, how can we account for the fact that irregulars usually take longer to compute than regulars? (Cf. Prasada, Pinker, Snyder, 1990). A possible solution suggested by Pinker and Prince (1991) is that the two processes might run in *parallel*. If we assume that the two (sub)modules operate in parallel, as soon as a cluster of irregular sounding phonemes is encountered in the associative templates module, the default rule module is turned off but the actual production of the irregular past tense occurs only after this event, resulting in a slower production of the irregular pasts; but if nothing interferes with the default-rules (sub)module, then the production is computed directly. Thus, when regular forms are computed, they are computed faster than irregular forms which require more processing steps. The relative reaction time of associate templates and rules can also be related to the frequency of their usage. This can explain the fact that more frequent irregular verbs are retrieved faster than less frequent irregular verbs.

One question that might arise in connection with our conjecture that the English past-tense inflection is controlled by two precedence-ordered (sub)modules is how these (sub)modules are related to the traditional distinction between lexicon and grammar. It is clear that the first-order default rule module which deals with the formation of the regular past should be placed "in the grammar", i.e. these first-order rules are directly interfaced with other grammatical rules including phrase-structure rules. However, we do not make any commitments about where the associative-templates (sub)module is situated. It could be a part of the lexicon in which case each irregular verb should be encoded together with its irregular template. On the other hand, the irregular (sub)module can as well be part of the grammar in which case each template is associated with more than one irregular verb. This later option is favoured by the fact that irregular-sounding pseudo verbs may elicit irregular past responses in human subjects.

6 Postscript: An Ultimate Comparison of the Raw Generalization Power

Brian MacWhinney informed us[8] that he had the testing results of the ANN simulation that were not published in M&L, and that his results were virtually identical to those of our SPA. We find that his recent ANN model is actually slightly different from the one in M&L. For example, one layer of 200 hidden units (instead of two layers of 200 units each) is used, no copy connections are built-in, and both input and output patterns (instead of just the input patterns) are encoded by left and right justified templates. MacWhinney informed us that 1,200 verbs were in the training set, among which 1081 were regular and 119 were irregular. Training took 4,200 epochs, and he got 100% correct on regulars and 80% on irregulars. The testing set consisted of 87 regulars and 15 irregulars. The percent correct on testing at epoch 4,200 was 91% for regulars and 27% for irregulars. His results looked indeed compatible with those of the SPA. However, the training and testing sets are different, therefore this is not a head-to-head comparison[9].

We have spent a considerable amount of time and effort trying to replicate his experiment, hoping to be able to apply our SPA on exactly the same set of verbs. We started with the backpropagation algorithm supplied in the book by McClelland and Rumelhart (1988). How-

[8]via personal communication. MacWhinney also left files of the ANN simulation in an ftp site poppy.psy.cmu.edu.

[9]Finally we realize (via private communication) that it is the representation in MacWhinney's recent connectionist implementation that improves his results. The input (for verb base) is coded by the right-justified template. The output contains two parts: a right-justified template same as the input, and a coda (VVCCC). The right-justified template in the output is used to represent the verb base without including the suffix for regular past tense. The suffix is represented in the coda. For irregular past tense, the coda is left empty. Such representation facilitates learning greatly. For regulars, the right-justified output is always identical to the input, verb ending is always at the same position, and the suffix is separated from the verb base. If our SPA is applied on this representation, the predictive accuracy will be improved further dramatically.

ever, this turned out to be a very frustrating process. *First* of all, the whole training process up to 4,200 epochs on a network of this size would take over 12 days (actual running time with a single job) on a powerful IBM RISC 6000 machine. *Second*, several attempts of setting the learning rate (0.1, 0.2, 0.5), the momentum (0.9, 0.25), and the number of hidden units (120, 200, 300) resulted in a failure to learn: the curve of the training error went almost flat except one or two sudden and small drops during several days of training. Learning was apparently struck in a local minimum. *Third*, we believe that to compare the true generalization ability of two learning models, one should use a relatively small set of training examples. If over 92% of the total data are used in training as in MacWhinney's recent simulation, SPA and ANN may indeed have very close performance since asymptotically both will reach very high accuracy in testing when the training set becomes larger. Using such a large training set also leaves a small testing sample so the result of the testing accuracy becomes less reliable. *Fourth*, we believe that only multiple runs of random samples of training and testing would yield reliable estimation of the true generalization ability of the learning models.

At that time we heard that an ANN package called Xerion developed in University of Toronto had several more sophisticated search mechanisms than the steepest gradient descent method with momentum. Using a conjugate gradient method and a line search technique, some of the same training sets on which training did not converge before converged with Xerion. (We noted that using steepest gradient search with momentum in Xerion still failed to converge.) From that point on we switched to Xerion which is installed on a SunSparc 10.

To have a direct, head-to-head comparison of the *raw generalization power* between the SPA and the ANN simulation, we decided to set up a more reliable test procedure as follows:

1. We used 500 verb pairs in the training sets, and the testing set was increased to 500 verb pairs as well. Training and testing sets were disjoint.

2. Random sampling was used to create 5 training/testing data sets as described in 1 above. Five runs of SPA and ANN were conducted and both SPA and ANN were trained and tested on the exact the same data set in each run.

3. No templates were used. The input and output format were simply 8 left to right phoneme letters. For example, the verb bet (in UNIBET coding as bEt) would be represented as bEt$$$$$ where $ stands for the blank phoneme. Both SPA and ANN took this format. Verb stems and their past tenses which have more than 8 phoneme letters were removed. This resulted in a total of 1130 verb pairs.

The *only* difference between SPA and ANN is that the SPA takes phoneme letters directly but ANN has to encode each phoneme letter to binary bits. Since the SPA does not use, explicitly or implicitly, any phonetic features in as in M&L, we decided not to use phonetic features in the ANN model as well in this ultimate comparison. (We performed further comparisons using templates and phonetic features as in M&L. See later in this section.) In fact, we could not use phonetic features anyway because the vowels and consonants have a different and disjoint set of phonetic features in M&L, but without the templates each position could be either vowels or consonants. Since there are total of 37 (36 plus the blank) phoneme letters, we used a *dense encoding scheme*: assign bit patterns of 6 bits ($2^6 = 64 > 37$) to each of the 37 phoneme letters randomly. Thus, the network consisted of $6 \times 8 = 48$ input bits and 48 output bits. We tried various numbers of hidden units and found that 100 hidden units work the best. To decode the network output back to phoneme letters, the phoneme letter that has the minimal Hamming distance with the network output is chosen. If the decoding process ties with several phoneme letters with the same minimal Hamming distance, one of them is chosen randomly. This directly reflects the probability of the correct decoding at the level of the phoneme letters. When all of the 8 phoneme letters are decoded, if one or more (including the blank phoneme letter $) are incorrect, the whole pattern is counted as incorrect at the word level.

Training with the SPA was always 100% correct since there was no contradiction in the data set[10]. Training of ANN (with conjungate gradient search) stopped when the search converged or the line search failed. The network errors (sum of the squared difference over all output bits and all training patterns) were usually quite small (from 22 to 35). The trained network thus had very high accuracy on the training set after decoding. In Table 14 we list the training results of ANN. As we can see, almost all regular verbs were trained correctly. This coincides with previous ANN simulations.

		SPA: % Correct on Training			ANN: % Correct on Training		
	Reg/Irreg.	Reg.	Irreg.	Combined	Reg.	Irreg.	Combined
Run 1	443/57	100.0	100.0	100.0	99.1	60.0	94.6
Run 2	444/56	100.0	100.0	100.0	98.9	80.4	96.8
Run 3	447/53	100.0	100.0	100.0	99.6	71.7	96.6
Run 4	454/46	100.0	100.0	100.0	99.6	63.0	96.2
Run 5	462/38	100.0	100.0	100.0	99.1	78.9	97.6

Table 14: Accuracy on training without phonetic feature encoding and without templates.

It is more important to compare how well verbs unseen in the training sets are generalized. Table 15 summarizes the results of the testing accuracies of SPA and ANN on testing set of 500 unseen verb pairs after decoding. From this table, it is very clear to see that the SPA has much better generalization abilities than the ANN simulation. The accuracy of the SPA on regular verbs is more than twice as high as the ANN model. Except in Run 5, the accuracy of the SPA on irregular verbs is also much higher. Overall, the combined accuracy of SPA is 2.28 times higher that of the ANN model.

[10]We only keep one form among multiple past tenses, such as *hang-hanged* and *hang-hung*, in the data set. Also no homophones exist in the data set.

		SPA % Correct on Testing			ANN % Correct on Testing		
	Reg/Irreg.	Reg.	Irreg.	Combined	Reg.	Irreg.	Combined
Run 1	449/51	83.1	9.8	75.6	39.4	2.0	35.6
Run 2	442/58	87.1	15.5	78.8	34.8	5.2	31.4
Run 3	444/56	83.1	8.9	74.8	31.1	5.4	28.2
Run 4	440/60	84.3	8.3	75.2	41.4	1.7	36.4
Run 5	435/65	86.7	7.7	76.4	40.0	6.2	35.4
Average		84.9	10.0	76.2	37.3	4.1	33.4

Table 15: Comparisons of testing accuracy of SPA and ANN (no templates and no phonetic feature encoding)

At this point we became very interested in knowing how templates and phonetic features as in M&L might help improve the generalization ability of the learning models. Recall that according to M&L, one part of the templates is left justified, and is in the form of CCCVVCC-CVVCCCVVCCC, where C stands for consonant and V for vowel space holders. In M&L(and MacWhinney's recent implementation of ANN), a right justified template in the form of VVCCC is also used for the same verb (c.f. Section 2.4). In addition, each phoneme letter is encoded by a set of (binary) phonetic features. Each vowel is encoded by 8 phonetic features (front, centre, back, high, low, middle, round, and diphthong) and each consonant by 10 phonetic features (voiced, labial, dental, palatal, velar, nasal, liquid, trill, fricative and interdental).

In our experimental comparison, we decided not to use the right justified template (VVCCC) since this information was redundant, and further, the left and right justified templates could contradict to each other in the output. Therefore, we only used the left justified template (CCCVVCCCVVCCCVVCCC) in both SPA and ANN. To ease the implementation, we added two extra features that were always assigned to 0 in the vowel phonetic feature set. Therefore both vowels and consonants were encoded by 10 binary bits. The ANN thus had $18 \times 10 = 180$ input bits and 180 output bits, and we found one layer of 200 hidden units (same as MacWhinney's new model) reached low training errors in 3 out of 5 runs, each with 500 verb pairs in both training and testing set. Two runs failed to converge with a small error.

The SPA is also trained with the binary bit patterns of 180 input bits and 180 output bits, exactly the same as the ones in the ANN simulation. To see how phonetic feature encoding affects the generalization, the SPA is also trained and tested on the same data sets but using phoneme letters directly. That is, 18 decision trees were built for each of the phoneme letters in templates CCCVVCCCVVCCCVVCCC. As before, training with SPA always reached 100.0% accuracy.

Testing accuracy on novel verbs came up with several exciting and surprising results. Because the training sets are smaller, the accuracy on testing is lower for both SPA and ANN.

However, smaller training sets make the difference in testing accuracy *more evident*. With the template and phonetic feature encoding, the ANN model and the SPA using binary bits have a very similar accuracy, while the SPA using phoneme letter encoding (still with the template) produces *higher* accuracy on testing. Comparing Table 16 and Table 15, we find that the SPA using phoneme letters directly performs slightly better without templates than the one with templates on regular verbs. The percent correct on regular verbs is conceptually more important than that on irregular verbs due to the reasons discussed in Section 4.4.2. The testing results of ANN and SPA can be found in Table 16.

	Template, feature encoding						Template, letter encoding		
	ANN: % Correct			SPA: % Correct			SPA: % Correct		
	Reg.	Irrg.	Comb.	Reg.	Irrg.	Comb.	Reg.	Irrg.	Comb.
Run 1	62.4	12.5	57.6	62.2	18.8	58.0	83.0	29.2	77.8
Run 2	56.3	6.9	50.6	57.9	8.2	52.2	83.3	22.4	76.2
Run 3	58.2	16.0	54.0	58.0	8.0	53.0	80.9	20.0	74.8
Average	59.0	11.8	54.1	59.4	11.7	54.4	82.4	23.9	76.3

Table 16: Comparisons of testing accuracy of SPA and ANN (with templates and phonetic feature encoding)

Our results seriously undermine the advantages of the distributed representation of artificial neural networks, a unique feature that connectionists have been advocating. We have demonstrated that in this task, the distributed representation is actually a bad thing. The SPA using templates performs slightly inferior to SPA without templates on regular verbs, and SPA (and ANN) with phonetic feature encoding performs much inferior to SPA with phoneme letter directly. The distributed representation does improve the ANN's generalization ability comparing to the ANN with the dense encoding; but the SPA with the direct phoneme representation and with the error-correcting codes outperforms the ANN *by a wide margin*. These results of our ultimate comparisons are summarised in Table 17. Our results challenge the view that the connectionist model with distributed representation is suitable for many aspects of human cognition.

BETTER $\Longleftarrow$	Generalization Abilities		$\Longrightarrow$ WORSE
SPA no templates phoneme letters	SPA templates phoneme letters	SPA, ANN templates phonetic features	ANN no templates dense codes

Table 17: Summary of the ultimate comparisons between SPA and ANN.

Is there any theoretical reason that our symbolic based learning algorithm SPA outperforms the ANNs on the task of learning past tenses of English verbs? We believe there are good reasons for it. In terms of learning discrete patterns, it can be shown that there exist some input/output patterns that symbolic methods learn and generalize qualitatively better than ANNs, and vise versa. This fact reflects the different inductive biases of the different learning algorithms. A *bias* is a preference of choosing certain hypotheses over others. Among all possible sets of production rules that fit the training sample, ID3 and SPA will induce simple decision trees (small sets of production rules) rather than complicated ones. On the other hand, the trained ANN can usually be characterized by a small set of hyperplanes. Whether a learning algorithm learns and generalizes well on a training set depends on how well the underlying regularities of the data set *"aligns"* with the bias of the learning algorithm. In other words, it depends on if the underlying regularities of the data set can be represented *compactly* in the representation used by the learning algorithm. If the underlying regularities can be represented by a simple decision tree, or a small set of production rules, then even with a small training sample the decision tree learning algorithms such as ID3 and SPA can "capture" these regularities and generalize very well. ANNs are usually more suited for functions represented by a small set of linear relations with numerical attributes. However, in Section 5 the underlying regularities governing the formation of the past tenses of English verbs form a small set of production rules when we use direct phoneme letter encoding. This is especially so in the case of regular verbs. That is why SPA generalizes very well. The SPA performs worse in the distributed representation than in the symbolic phoneme letter encoding directly because rules with 180 input attributes are more complicated than with phomemes, and the SPA is faced with a harder time to pick up the most informative one in the decision tree building process, let alone each attribute is actually much less informative. To summarize, symbolic rule/decision tree learning systems such as our SPA should outperform ANNs on learning past tense of English verbs.

Finally, we note that the SPA is much faster to learn than the ANN simulation. For the phonetic feature encoding with templates (i.e. both input and output use 180 binary attributes), training 500 patterns on ANN takes over 100,000 CPU seconds (over 27 hours) on a SunSparc 10, but it takes less than 1,000 CPU seconds (16 minutes) on the same machine using SPA. Therefore, the SPA is over 100 times faster than ANNs.

7 Conclusion

In summary, we established the following results:

First, we demonstrated that the eliminative connectionists's claim that there are no symbolic models capable of matching or surpassing the performance of artificial neural networks in the learning of the past tenses of English verbs is *no longer valid*. The SPA has significantly outperformed Rumelhart and McClelland's as well as MacWhinney and Leinbach's models on this task.

Second, we argued that in principle for pattern association tasks where the underlying regularities are governed by a small set of production rules, symbolic rule-induction learning models such as SPA generalize better than ANNs. connectionist models have *no magic power* and the distributed representation can actually be a bad thing in solving these problems.

Third, we demonstrated that in sharp contrast to ANNs which are unable to represent knowledge in explicit form, the SPA can produce *explicit representations* in the form of production rules. We also argued that the ability to represent knowledge explicitly is indispensable in solving the full induction problem of learning the past tenses of the English verbs.

In light of these findings, we conclude that eliminative connectionists' vision of cognition as pattern association and pattern recognition without explicit symbolic representation is inadequate. Pattern association as such need not imply rule-less or cue-based models of language acquisition or human learning in general. If there are any production rule-like regularities in the sets of patterns, they will be extracted and represented in symbolic form, ready for further processing. In contrast, the rival connectionist models do not offer any form of knowledge representation and leave the further processing and integration of the acquired knowledge a complete mystery. Because of this the eliminativist models cannot explain how mature speakers acquire their knowledge of the language. We hope that our theoretical and experimental comparison between the SPA and the rival connectionist models is sufficient to show that the rule-based symbolic approach stands a far better chance in explaining language learning, language processing, and cognition in general than eliminative connectionism.

Acknowledgement

The authors gratefully thank Steve Pinker for his constant encouragement and his very detailed comments on the early draft of the paper. We thank Brian MacWhinney for providing the verb data used in their simulation. We also thank Steve Cherwenka for helping implement the SPA.

References

Aronoff, M. (1976). Word Formation in Generative grammar. Cambridge, MA: MIT Press.

Atlas, L, Cole, R., Connor, J., El-Sharkawi, M., Marks, R., Muthasamy, Y., & Barnard, E. (1990). Performance comparisons between backpropagation networks and classification trees on three real-world applications. In D. Touretzky, (Ed.), Advances in Neural Information Processing Systems (pp. 622- 29), Vol. 2, San Mateo, CA: Morgan Kaufmann Inc.

Bybee, J. & Slobin, D. (1982). Why small children cannot change language on their own: Suggestions from the English past tense. In A. Ahlqvist (Ed.), Papers from the 5th International Conference on Historical Linguistics. (Current Issues in Linguistic Theory, Vol. 21) Philadelphia/Amsterdam: Benjamins.

Churchland, P.M. (1988). Matter and Consciousness. Revised edition. Cambridge, MA : MIT Press.

Churchland, P.M. (1989). A Neurocomputational Perspective. Cambridge, MA : MIT Press.

Egedi, D. and Sproat, R. (1991). Connectionist networks and natural language morphology. Unpublished manuscript. Linguistics Research Department, AT&T Bell Lab., Murray Hill, NJ.

Feng, C., King, R., Sutherland, A., & Henery, R. (1992). Comparison of symbolic, statistical and neural network classifiers. Manuscript. Department of Computer Science, University of Ottawa.

Fisher, D. & McKusick, K. (1989). An empirical comparison of ID3 and back-propagation. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (pp. 788-793). Detroit, MI.

Fodor, J.A. & McLaughlin, B.P. (1990). Connectionism and the problem of systematicity. Cognition, 35, 183-204.

Fodor, J. A. & Pylyshyn, Z. (1988). Connectionism and Cognitive Architecture: A Critical Analysis. In S. Pinker & J. Mehler, (Eds.), Connections and Symbols (pp. 3-71). Cambridge, MA : MIT Press.

Gorman, R.P. & Sejnowski, T.J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. Neural Network, 1 (Part 1):75-89.

Hunt, E., Marin, J., & Stone, P. (1966). Experiments in Induction. New York, NY: Academic Press.

Kim, J., Pinker, S., Prince, A., and Prasada, S. (1991). Why no mere mortal has ever flown out to center field. Cognitive Science, 15, 173-218.

Kiparsky, P. (1982). Lexical phonology and morphology. In I.S.Yang (Ed.), Linguistics in the Morning Calm, (3-91). Seul: Hansin

Lachter, J. & Bever, T. (1988). The relation between linguistic structure and associative theories of language learning – A constructive critique of some connectionist learning models. In S. Pinker & J. Mehler, (Eds.), Connections and Symbols (pp. 195-247). Cambridge, MA : MIT Press.

Lapointe, S., Ling, C.X., & Matwin, M. (1993). Constructive Inductive Logic Programming. In Proceedings of IJCAI-93 (Thirteenth International Conference on Artificial Intelligence). Morgan Kaufmann Publishers.

Ling, C.X. (1994). Learning the past tense of English verbs: the Symbolic Pattern Associator vs. connectionist models. Journal of Artificial Intelligence Research. To appear.

Marcus, G., Pinker, S., Ullman, M., Hollander, M., Rosen, T., and Xu, F. (1993). Overregularization in language acquisition. Monographs of the Society for Research in Child Development, 57(4, Serial No. 228).

McClelland, J. & Rumelhart, D. (1988). Explorations in Parallel Distributed Processing. The MIT Press.

McClelland, J., Rumelhart, D., & Hinton, G. (1986). The appeal of parallel distributed processing. In Rumelhart, D., McClelland, J. & the PDP Research Group (Eds.), Parallel Distributed Processing, Vol. 1 (pp. 3-44). Cambridge, MA: MIT Press.

MacWhinney, B. (1990). The CHILDES Project: Tools for Analyzing Talk. Hillsdale, NJ: Erlbaum.

MacWhinney, B. & Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb model. Cognition, 40, 121-157.

Marinov, M. (1993). On the spuriousness of the symbolic/subsymbolic distinction. Minds and Machines (forthcoming).

Muggleton, S. & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. Proceedings of the Fifth International Conference on Machine Learning, Morgan Kaufmann.

Pinker, S. (1984). Language Learnability and Language Development, Cambridge, MA: Harvard University Press

Pinker, S. (1991). Rules of Language. Science, 253, 530-535.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. In S. Pinker & J. Mehler, (Eds.), Connections and Symbols (pp. 73-193). Cambridge, MA : MIT Press.

Pinker, S. and Prince, A. (1991). Regular and irregular morphology and the psychological status of rules of grammar. In L. Sutton (Ed.) Proceedings of the 1991 Meeting of the Berkeley Linguistic Society. Berkeley, California: Berkeley Linguistics Society.

Plotkin, G. (1970). A note on inductive generalizations. In B. Meltzer & D. Michie, (Eds.), Machine Intelligence, Vol. 5, (pp.153-63). New York: North Holland.

Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. Cognition, 38, 43-102.

Pollack, J. (1991). Recursive distributed representations. In G.E. Hinton, (Ed.), Connectionist Symbol Processing (pp. 77-106). Cambridge, MA: MIT Press.

Prasada, S. and Pinker, S. (1993). Generalization of regular and irregular morphological patterns. Language and Cognitive Processes. 8(1), 1–56.

Quinlan, R. (1986a). Induction of decision trees. Machine Learning, Vol. 1, pp. 81-106.

Quinlan, R. (1986b). Probabilistic decision trees. In Y. Kodratoff, & R. Michalski, (Eds.), Machine Learning: An Artificial Intelligence Approach, Vol. 3, (pp. 140-152). San Mateo, CA, USA: Morgan Kaufmann, Inc.

Quinlan, R. (1989). Unknown attribute values in induction. In B. Spatz, (Ed.), Proceedings of the Sixth International Workshop on Machine Learning, San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., pp. 164-68.

Ripley, B.D. (1992). Statistical aspects of neural networks. In invited talk at SemSat, Sandjerg, Denmark, April 1992. Chapman and Hall.

Rumelhart, D. & McClelland, J. (1986). On learning the past tenses of English verbs. In Rumelhart, D., McClelland, J. & the PDP Research Group (Eds.), Parallel Distributed Processing, Vol. 2 (pp. 216-271). Cambridge, MA: MIT Press.

Rumelhart, D. & McClelland, J. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing? In B. MacWhinney (Ed.), Mechanisms of Language Acquisition. Hillsdale, NJ: Erlbaum.

Sejnowski, T. & Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. Complex Systems, 1, 145-168.

Shavlik, J., Mooney, R., & Towell, G. (1991). Symbolic and neural learning algorithms: An experimental comparison. Machine Learning, 6, 111-143.

Slobin, D.I. (1971). On the learning of morphological rules: A reply to Palermo and Eberhart. In D.I. Slobin (Ed.), The ontogenesis of grammar: A theoretical symposium. New York:

Academic Press.

Smolensky, P. (1988). On the proper treatment of connectionism. Behavioral and Brain Sciences, 11, 1-74.

Smolensky, P. (1991a). Connectionism, constituency, and the language of thought. In B. Loewer & G. Ray, (Eds.), Meaning and Mind: Fodor and His Critics (pp.201-228). Cambridge, MA: Basil Blackwell.

Smolensky, P. (1991b). Tensor product variable binding and the representation of symbolic structures in connectionist systems. In G.E. Hinton (Ed.), Connectionist Symbolic Processing (pp. 159-216). Cambridge, MA: MIT Press.

Thrun, S.B. at el. (1991). The MONK's problems - a performance comparison of different learning algorithms. Technical Report, Computer Science Department, Carnegie Mellon University.

Touretzky, D. (1991). BoltzCONS: Dynamic symbol structures in a connectionist network. In G.E. Hinton (Ed.), Connectionist Symbolic Processing (pp. 5-46). Cambridge, MA: MIT Press.

Tsaptsinos, D., Mirzai, A., & Jervis, B. (1990). Comparison of machine learning paradigms in a classification task. In G. Rzevsk, editor, Applications of artificial intelligence in engineering V: Proceedings of the fifth international conference. Springer-Verlag.

Weiss, S.M. & Kulikowski, C.A. (1991). Computer Systems that Learn: classification and prediction methods from statistics, neural networks, machine learning, and expert systems. Morgan Kaufmann, San Mateo, CA.