Issues in Comparing Symbolic and Connectionist Models

Charles X. Ling
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
E-mail: ling@csd.uwo.ca
WWW: http://www.csd.uwo.ca/faculty/ling

Abstract

There has been a heated debate between connectionist and symbolic models on the task of learning the past tense of English verbs. Claims are often made, but not often justified, that a new model has a superior generalization ability to the previous ones. In this paper, we first set up a proper criterion for making comparisons between models. We point out a crucial issue in comparison which has been largely ignored in the past. Then we present results on the generalization ability of the symbolic pattern associator, SPA. We challenge connectionist researchers to design connectionist models with similar or better generalization ability.

Introduction

Learning the past tense of English verbs, a minor aspect of language acquisition and processing, has received extensive study in the last few years. In 1986, Rumelhart and McClelland (1986) first designed and implemented a connectionist model of past-tense acquisition. Claims were made that such connectionist models, while requiring no symbol processing, grammatical rules, or explicit representation as in the traditional grammatical theories, are better models for past-tense acquisition and for language acquisition in general. Over the years, a number of criticisms of connectionist modeling appeared (Pinker & Prince, 1988; Lachter & Bever, 1988; Prasada & Pinker, 1993; Ling & Marinov, 1993), and there has been a heated debate over the symbolic and connectionist modeling of the task. Several subsequent attempts at improving the original results with new connectionist models have been made (Plunkett & Marchman, 1991; Cottrell & Plunkett, 1991; MacWhinney & Leinbach, 1991; Daugherty & Seidenberg, 1993). On the other hand, several symbolic models have been built to demonstrate their superior generalization abilities on the same task (Ling & Marinov, 1993; Ling, 1994; Mooney & Califf, 1995). However, it seems quite possible that a better connectionist model (or symbolic model) can always be constructed to outperform the previous counterpart on the generalization accuracy.

In this paper, we attempt to set out some criteria for a proper comparison between competing models. We will also discuss a crucial issue which has been largely ignored in the previous modeling of past tense acquisition.

A Proper Comparison Criterion

A computational model of any learning task consists of many components: the learning algorithm (and its parameters), data sampling method, training regime (batch or on-line, order of presentation), and representation format of the data (set of attributes). Each component can affect dramatically the learning behavior of the model. Clearly, to compare two different learning models on the same task, we must keep all components which are common to the two models constant. Such a practice has been used widely in the machine learning community. Result of comparison based on different representations, for example, is not very meaningful.

However, one complication in comparing different learning models is that the components in models may not be independent. For example, symbolic learning algorithms can use multiple-valued discrete attributes directly, while connectionist models normally take distributed representation. When this occurs, we should allow model-specific components to be different, and to be optimized for the learning behavior of the model. For example, while keeping everything else the same, an optimal distributed representation should be used in connectionist models. In addition, algorithm's parameters are model specific, and they should be carefully chosen to optimize its performance.

The same criterion should be applied to the structure of different models. The paradigm (or general structure) of the symbolic and connectionist models in comparison must match. This issue is further discussed in the following section.

SPA and Corresponding Connectionist Models

We use SPA (Ling & Marinov, 1993; Ling, 1994) as the symbolic model for the past tense learning. SPA is a general-purpose N-to-M (N input attributes to M output attributes) pattern associator which essentially applies the N-to-1 decision-tree classifier ID3 (Quinlan, 1986) M times. Given a set of patterns containing N input attributes and M output attributes, M decision trees are built by the SPA, one for each output attribute. With each tree determining one attribute value in the output, M trees will collectively predict the whole output pattern. For the verb past-tense learning, the input patterns are phoneme letters in the verb base, with a maximum of N phoneme letters. Similarly, the output patterns are phoneme letters in the past tense, with a maximum of M phoneme letters. For details of SPA, also see (Ling & Marinov, 1993, pages 248–255).

In general, SPA is similar in structure to layered, feed-forward connectionist models. However, several variations exist. Different SPA architectures should be matched to different connectionist architectures. In comparison, a matching architecture must be chosen, other-

wise the superiority of a model may not be claimed since it may compare to a "strawman" of another model.

There are three structures (versions) of SPA which have their correspondences in connectionist models. In the first version of SPA, the original C4.5 is called in constructing decision trees for output attributes. This version of SPA with the original C4.5 is equivalent in structure to the feedforward, layered connectionist models, with links connecting between layers of units.

In the original C4.5, however, when classifying a new example after the decision tree is built, if the new example falls into an empty leaf where no training examples have fallen, C4.5 would use the majority class in that branch as the class. We call this strategy *majority default strategy*. This strategy is clearly not ideal for learning past tense, since the stem of a regular verb is always "copied" into the past tense, rather than taking the most popular letter from other regular verbs.

In an improved version of SPA, an *adaptive default strategy* is implemented (Ling, 1994). The idea came from MacWhinney and Leinbach (1991): their connectionist models have "copy" links, which connect directly from the input units to the output units. This facilitates the identify copy of verb stem into past tense. However, the suffix of the past tense should not be copied, thus, such copy links also compete with links connecting between layers. Similarly in this SPA, there is also a competition between the "copy default strategy" and the "majority default strategy" (thus, we call it the adaptive default strategy). Basically, if a testing example falls in a leaf which is empty, SPA decides which default strategy to take. If more examples in this branch use the copy strategy than the majority strategy, then the copy strategy is used for this testing example; otherwise, the majority strategy is employed.

The version of SPA with adaptive default is equivalent in structure to feedforward connectionist models with direct connections between input and output units.

In the third version of SPA (which is really a representation change), an N-to-N pattern mapping problem is changed to an N-to-1 classification problem by using a moving window, as used in NETtalk (Sejnowski & Rosenberg, 1987). Basically, a window of a certain width moves from left to right to the N input attributes. At each time, the output attribute at the center of the window is learned and predicted, using input attributes currently in the window. This representation format effectively makes the specific position of linearly ordered attributes irrelevant: the regularity is learned according to the attribute in the center of the window and its left and right neighbours. For the verb past tense, instead of learning regular suffixation at different positions (and thus reducing the training set on each position), regular suffixation is learned at one position: the center of the window. In addition, it can deal with verbs of any length.

This version of SPA with moving window is equivalent in structure to feedforward connectionist models with moving window representation, or the recurrent connectionist models (Elman, 1990), which uses recurrent links and recurrent units to memorize the attributes outside the input attributes. Recurrent networks have been used in past tense acquisition previously (Cottrell & Plunkett, 1991).

When comparing to SPA, a connectionist model whose structure is equivalent to SPA should be chosen. That is, results from SPA with the majority default strategy should be matched to the ones from feedforward networks, results from SPA with the adaptive default strategy should be matched to the ones from feedforward networks with direct links from input to output units, and results from SPA with moving window should be matched to the ones from recurrent networks or feedforward networks with moving window.

What to Compare

Most computational models of past tense learning focus on both their generalization ability (predictive accuracies on unseen verbs), and on psych-linguistic behaviors of the models compared to humans. We will discuss the generalization ability of the models in this paper, and an important issue of generalization which has been largely ignored in previous models.

In many previous models on past tense learning, only a handful verbs were removes (sometimes hand-picked) from the training set for the testing purpose, and only one run was made. Such results were not reliable for several reasons: First, connectionist models (as well as decision-tree symbolic models) are computationally powerful. With a proper network architecture and enough training examples, they can represent any arbitrarily complex mappings. Thus, with a very large training set, the predictive accuracy of such models tends to be "saturated", and the difference in predictive accuracies of different models tends to be minimized. Therefore, to boost the difference between two models in comparison, one must use relatively small training sets. It would also be useful to compare model's behavior with that of human on small training sets. Therefore, a learning curve which reflects testing accuracies for training sets with very small to large sizes is crucial for comparing models.

The second problem is that when the testing set is too small, the testing accuracy is not reliable. Multiple runs should be performed to get averaged results. Finally, sampling of training and test sets should be done randomly. This eliminates human interference and improves the reliability of the result.

In this paper, we train SPA on randomly sampled training sets of different sizes to produce learning curves under various settings.

SPA Results

We present SPA's results on the generalization ability in this section. The datasets used can be downloaded anonymously from the author's web site. Together with SPA programs at the same site, the results reported here can be replicated easily by other researchers.

Representation Format

Our verb set came from MacWhinney's original list of verbs. The set contains about 1400 verb stem and past tense pairs. Learning is based upon the phonological UNIBET representation (MacWhinney, 1990), in which different phonemes are represented by different alphabetic and numerical letters. There is a total of 36 phonemes. For example, Table 1 lists several verbs in spelling form, and in different UNIBET representation formats (see Section for more details). Pairs of verb stem and past tense in UNIBET representation are used in the learning tasks.

Table 1: Different representation formats of several sample verbs and their past tenses.

Spelling	UNIBET	Template CCCVVCCCVVCCCVVCCC	Left-justified PPPPPPPPPPPPPPP	Right-justified with Coda PPPPPPPPPPPPPPP VC
abandon	6b&nd6n	___6_b__&_nd_6_n__	6b&nd6n_____	_____6b&nd6n
abandoned	6b&nd6nd	___6_b__&_nd_6_nd_	6b&nd6nd_____	_____6b&nd6n _d
blend	blEnd	bl_E_nd_____	blEnd_____	_____blEnd
blended	blEndId	bl_E_nd_I_d_____	blEndId_____	_____blEnd Id
attack	6t&k	___6_t__&_k_____	6t&k_____	_____6t&k
attacked	6t&kt	___6_t__&_kt_____	6t&kt_____	_____6t&k _t
arise	6r3z	___6_r__3_z_____	6r3z_____	_____6r3z
arose	6roz	___6_r__o_z_____	6roz_____	_____6roz __
become	bIk6m	b__I_k__6_m_____	bIk6m_____	_____bIk6m
became	bIkem	b__I_k__e_m_____	bIkem_____	_____bIkem __

In the past, several different representation formats have been used in learning. Here, we have chosen the three most popular ones.

Template Representation This representation is suggested by MacWhinney and Leinbach (1991). Both input and output patterns are fitted in templates in the format of CCCVVCCCVVCCCVVCCC, where C stands for consonant and V for vowel space holders. A similar template, CCCVVCCC, was used in Daugherty and Hare (1993)'s connectionist model. The idea behind this representation is to align consonants and vowels so that the ending letter of regular verbs and similarity patterns in certain irregular verbs are in more deterministic positions. See examples in Table 1.

Left-Justified Representation This representation is straightforward left-justified phoneme letters with a total length of 15 (no verb or past tense has more than 15 phonemes). We use PPPPPPPPPPPPPPP to denote such representation, where P is a phoneme. See Table 1 for examples of the left-justified representation.

Right-justified Representation with Coda In (MacWhinney, 1993), a new representation is used. The input (for the verb stem) is coded by the right-justified template CCCVVCCCVVCCCVVCCC; the output contains two parts, a right-justified template that is the same as the one in the input, and a coda in the form of VC. The right-justified template in the output is used to represent the past tense *without* including the suffix for the regular verbs. The suffix of the regular past tense occurs in (and only in) the coda. For the irregular past tense, the coda is left empty. However, as we will see later, templates really do not help in generalization, so we simply use non-templated right-justified representation in the main part. See Table 1 for examples.

It is expected that such data representation facilitates learning. For the regular verbs, the main output patterns are always identical to the input patterns, and the verb-ending phoneme letter always appear as the last letter of the right-justified part. Because irregular past tenses have an empty coda, it allows learning algorithms to distinguish regular verbs from irregular verbs in the training set.

Since SPA can take symbolic attributes directly as input, SPA is applied to the phoneme letters directly for all of those representation formats. For the template representation, 18 decision trees were built for each phoneme letter in the output templates, taking input from 18 templated input phoneme letters. For the left-justified representation, 15 decision trees were built for each phoneme letter in the output, taking input from 15 left-justified input phoneme letters. For the right-justified representation, 17 (15+2) decision trees were built for each phoneme letter in the output, taking input from 15 right-justified input phoneme letters. For SPA with moving window, only one decision tree is built, and this tree is used for predicting each output phoneme in the past tense.

Learning both Regular and Irregular Verbs

From a whole set of about 1,400 regular and irregular verbs, we randomly sample training set of various sizes without replacement, and use the rest of the verbs as the testing sets. Thus, training and testing sets are disjoint, and test sets are maximized for more reliable results. Three different representation formats (template, left-justified, right-justified) are used in the experiment. For each, SPA with three different architectures (majority default strategy, adaptive default strategy, and moving window) are tested. For each training size, representation format, and architecture, 10 random samples are made, and predictive accuracies are averaged and recorded in Table 2.

From the table, several interesting conclusions can be drawn. First, comparing vertically, we see that the template representation is not particularly beneficial in learning, since the testing accuracies are comparable to the left-justified representation. The main reason for this is that the verbs have various lengths, so the endings of regular verbs are still distributed in various places in the templates, and need to be learned separately. However, the template makes it hard to locate the ending phoneme in the verb stem: seeing an empty space (with a non-empty left neighbor) does not imply that the verb has come to an end, while in the left-justified representation, it can. This is also why the results of SPA with moving window on the template representation are very bad.

Second, the right-justified representation often has the best testing accuracies. This is expectable, because the last phoneme of verb stems is located at the fixed location. Predictive accuracies for regular verbs are often much improved.

Third, horizontally, the SPA with the adaptive default strategy outperforms SPA with C4.5's majority default

Table 2: Testing accuracies of SPA on regular and irregular verbs with various training sizes (50, 100, 500, and 1000).

Testing accuracies of SPA with majority default strategy (Equivalent to feedforward networks)									
	Template			Left-justified			Right-justified		
Dataset	Reg	Irrg	Comb	Reg	Irrg	Comb	Reg	Irrg	Comb
50	23.9	6.5	22.1	20.7	4.1	19.0	32.9	4.1	29.9
100	51.6	6.8	47.0	48.0	5.5	43.6	57.5	3.8	52.0
500	79.0	14.5	72.4	80.4	13.4	73.6	82.2	13.2	75.2
1000	82.3	23.5	76.3	84.5	18.5	77.4	87.1	20.0	80.0

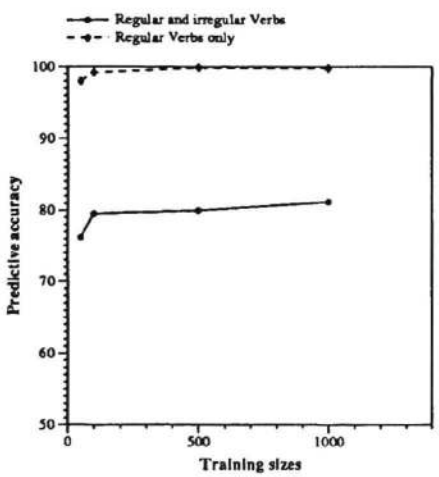
Testing accuracies of SPA with adaptive default strategy (Equivalent to feedforward networks with direct links from input to output)									
	Template			Left-justified			Right-justified		
Dataset	Reg	Irrg	Comb	Reg	Irrg	Comb	Reg	Irrg	Comb
50	40.7	9.4	37.4	42.4	5.1	38.5	82.3	5.8	74.4
100	63.1	7.9	57.4	65.0	6.4	58.9	87.9	4.4	79.3
500	81.1	14.7	74.3	83.2	13.4	76.2	88.4	14.1	80.9
1000	83.1	23.5	76.7	85.3	18.5	78.2	89.5	20.5	82.2

Testing accuracies of SPA with moving window (Equivalent to recurrent networks)									
	Template			Left-justified			Right-justified		
Dataset	Reg	Irrg	Comb	Reg	Irrg	Comb	Reg	Irrg	Comb
50	20.4	7.1	19.0	79.7	6.3	72.1	84.4	4.6	76.2
100	31.9	8.9	29.6	86.6	5.9	78.3	88.3	3.3	79.5
500	61.5	16.0	56.4	89.0	14.1	81.4	87.8	11.6	80.0
1000	73.9	24.2	68.6	89.3	16.3	81.6	88.8	16.6	81.2

strategy. In addition, SPA with moving window produces the best results (except for the template representation, for the reason discussed earlier).

Last, it is evident that even with 50 verbs in the training set, the overall predictive accuracy on unseen regular and irregular verbs is quite high: with the right-justified, moving window representation, it reaches 76.2%. We plot the learning curve of SPA with the right-justified, moving window representation in Figure 1.

Figure 1: Learning curve of predicting unseen testing verbs.



Learning Only Regular Verbs

Predicting the past tense of an unseen verb, which can be either regular or irregular, is not an easy task. Irregular verbs are not learned by rote as traditionally thought, since children and adults occasionally extend irregular inflection to irregular-sounding regular verbs or pseudo verbs (such as *cleef — cleft*) (Prasada & Pinker, 1993). Pinker (1991) and Prasada and Pinker (1993) argue that regular past tenses are governed by rules, while irregulars may be generated by the associated memory, which has this graded effect of irregular past-tense generalization. It would be interesting, therefore, to compare SPA and connectionist models on the past-tense generalization of regular verbs only. This may not be a psychologically plausible experiment, but the purpose here is to compare the generalization ability of the two models.

Note that even if one uses template or left-justified representations, learning regular past tenses is not as easy as one might think. Since regular verbs vary in length, regular inflection requires learning suffixation rules at *different* positions. In addition, instead of "just add ed" (as in the spelling), there are three different suffixes for regular verbs in phonological form. When the verb stem ends with *t* or *d* (UNIBET phonetic representations), then the suffix is *Id*. For example, *extend — extended* (in spelling form). When the verb stem ends with an unvoiced consonant, the suffix is *t*. For example, *talk — talked*. When the verb stem ends with a voiced consonant or vowel, the suffix is *d*. For example, *solve — solved*. More examples can be found in Table 1.

We used the same training sizes, representation formats, and SPA architectures as in the previous section in testing regular verbs. The testing accuracies averaged over 10 runs can be found in Table 3.

Clearly, the same conclusions can be drawn for regular verbs as in the last section. Further, it is clear that learning regular verbs only is much easier. The predictive accuracies are very high (over 98%), even with 50 regular verbs in training sets, when SPA with moving window is applied to the right-justified representation.

Table 3: Testing accuracies of SPA on regular verbs with various training sizes (50, 100, 500, and 1000).

SPA with majority default strategy (Equivalent to feedforward networks)			
	Template	L-justified	R-justified
Dataset	Reg	Reg	Reg
Reg 50	33.4	26.2	40.9
Reg 100	60.8	52.1	64.2
Reg 500	88.6	92.1	93.4
Reg 1000	92.8	96.9	97.2

SPA with adaptive default strategy (Equivalent to feedforward networks with direct links from input to output)			
	Template	L-justified	R-justified
Dataset	Reg	Reg	Reg
Reg 50	53.9	54.4	95.4
Reg 100	72.4	71.0	97.4
Reg 500	90.7	94.3	99.8
Reg 1000	93.8	97.7	99.8

SPA with moving window (Equivalent to recurrent networks)			
	Template	L-justified	R-justified
Dataset	Reg	Reg	Reg
Reg 50	22.2	88.7	98.0
Reg 100	37.3	95.3	99.2
Reg 500	73.3	99.2	99.9
Reg 1000	82.9	99.7	99.8

Again, we plot the learning curve of SPA with the right-justified, moving window representation in Figure 1.

A Challenge for Better Connectionist Models

As discussed in earlier, when comparing two learning models, we should keep the components common to the two models the same. We have obtained SPA's predictive accuracies with various training sizes, representation formats, as well as structures of the model. Results from connectionist models should be obtained under the same setting to make the comparison meaningful. In another word, results from connectionist models with a very different setting cannot be compared to the ones reported here.

In the previous publication, only few published results from connectionist models can be compared head-to-head to the ones in this paper. From Table 2, the average testing accuracy of SPA, with the majority default strategy and 500 training examples in the template representation, is 72.4%. With the same setting, Ling (1994, Page 222) reported that the average testing accuracy of the corresponding connectionist model (his implementation) is 56.6%. From Table 3, the average testing accuracies of SPA, with the majority and adaptive default strategies and 50 training verbs represented in template, are 33.4% and 53.9% respectively, while the corresponding connectionist models only 7.3% and 14.6% respectively (Ling, 1994, page 223). The difference is quite significant.

However, we did not run extensive experiments on connectionist models in this paper, because there are so many detailed, model-specific design choices and parameters in modeling, that such experiments should be best performed by experts in the area. Some of these design choices include the number of hidden layers, the number of units in each layer, the distributed representation, the learning algorithm and its parameters, encoding and decoding methods, and overfitting controls.

Therefore, we pose a challenge to connectionists to construct connectionist models with similar or better generalization accuracies under the same setting common to both models.

Conclusions

In this paper, we outlined a proper criterion to conduct a fair and reliable comparison between connectionist and symbolic models. We also suggest that the learning curve of predictive accuracies with various training sizes should be regarded as a crucial issue in model comparison. If a model is claimed to have a better generalization ability, its superior predictive accuracy on small training sets is particularly salient.

We then presented results from SPA under various commonly used representation formats and structures. Each structure of SPA corresponds to a certain architecture of connectionist models. All of the datasets used and the SPA programs can be accessed on-line. We hope that connectionist researchers can take the data and design connectionist networks with better generalization accuracies under the same setting common to both models.

Reference

Cottrell, G., & Plunkett, K. (1991). Using a recurrent net to learn the past tense. In *Proceedings of the Cognitive Science Society Conference.*

Daugherty, K., & Hare, M. (1993). What's in a rule? the past tense by some other name might be called a connectionist net. In *Proceedings of 1993 Connectionist Model Summer School.*

Daugherty, K., & Seidenberg, M. (1993). Beyond rules and exceptions: A connectionist modeling approach to inflectional morphology. In Lima, S. (Ed.), *The Reality of Linguistic Rules.* John Benjamins.

Elman, J. (1990). Finding structure in time. *Cognitive Science, 14,* 179–211.

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 Pinker, S., & Mehler, J. (Eds.), *Connections and Symbols,* pp. 195 – 247. Cambridge, MA: MIT Press.

Ling, C. X. (1994). Learning the past tense of English verbs: the Symbolic Pattern Associator vs. connectionist models. *Journal of Artificial Intelligence Research, 1,* 209 – 229.

Ling, C. X., & Marinov, M. (1993). Answering the connectionist challenge: a symbolic model of learning the past tense of English verbs. *Cognition, 49*(3), 235–290.

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

MacWhinney, B. (1993). Connections and symbols: closing the gap. *Cognition, 49*(3), 291–296.

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

Mooney, R., & Califf, M. (1995). Induction of first-order decision lists: Results on learning the past tense of english verbs. *Journal of Artificial Intelligence Research*, *3*, 1–24.

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 Pinker, S., & Mehler, J. (Eds.), *Connections and Symbols*, pp. 73 – 193. Cambridge, MA: MIT Press.

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

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

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81 – 106.

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.

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