Hybrid Computational Model for Producing English Past Tense Verbs

Maitrei Kohli[1] , George D. Magoulas[1], Michael Thomas[2]

[1]Department of Computer Science and Information Systems,
[2]Department of Psychological Sciences,
Birkbeck College, University of London, UK
{maitrei, gmagoulas}@dcs.bbk.ac.uk
{m.thomas}@psychology. bbk.ac.uk

Abstract. In this work, we explore the use of artificial neural networks (ANN) as computational models for producing English past tense verbs by combining them with the genetic algorithms (GA). The principal focus was to model the population variability exhibited by children in learning the past tense. This variability stems from genetic and environmental origins. We simulated the effects of genetic influences via variations in the neuro computational parameters of the ANNs, and the effects of environmental influences via a filter applied to the training set, implementing variation in the information available to the child produced by, for example, differences in socio-economic status. In the model, GA served two main purposes - to create the population of artificial neural networks and to encode the neuro computational parameters of the ANN into the genome. English past tense provides an interesting training domain in that it comprises a set of quasi-regular mappings. English verbs are of two types, regular verbs and the irregular verbs. However, a similarity gradient also exists between these two classes. We consider the performance of the combination of ANN and GA under a range of metrics. Our tests produced encouraging results as to the utility of this method, and a foundation for future work in using a computational framework to capture population-level variability

Keywords: Feed forward neural networks, imbalanced datasets, hamming distance, nearest neighbour, genetic algorithms, English past tense verbs, quasi regular mappings.

1 Introduction

Artificial neural networks (ANNs) are computational abstractions of the biological information processing system. In this work, we combine ANNs with genetic algorithms (GA) to develop a new computational model for learning English past tense verbs.

The English past tense has been widely studied as a testing ground for theories of language development. This is because it is quasi-regular mapping, comprising both a productive rule (add –ed to the verb stem to produce the past tense) and a set of

exceptions to this rule. This raises the question of the processing structures necessary to acquire the domain. Until now substantial amount of work has used ANNs as cognitive models of the acquisition process (see [1] for review). No work to date, however, has considered how to capture the wide range of variability that children exhibit in acquiring this aspect of language. Since ANNs constitute parameterised learning systems, they provide a promising framework within which to study this phenomenon [2].

Factors affecting language development are attributed to genetic and environmental influences. To model genetic influences, we use GA as a means to encode variation in the neuro computational parameters of the ANNs, thereby modulating their learning efficiency. These parameters are responsible for how the network is built (e.g., number of hidden units), its processing dynamics (steepness of the activation function), how it is maintained (weight decay), and how it adapts (learning rate, momentum). To model environmental influences, we apply a filter to the training set to alter the quality of the information available to the learning system. One candidate causal factor in producing environmental variation is socio-economic-status (SES). A body of research suggests that children in lower SES families experience substantially less language input and also a narrower variety of words and sentence structures [3]. The filter creates a unique subsample of the training set for each simulated individual, based on their SES.

This paper is organised as follows. First we provide information about the problem domain. Then the methodology adopted in our approach is described. Next we discuss the datasets used. In section 4 the experimental setups and performance assessment techniques are described. Finally we present the experimental results and discuss the findings.

2. The English past tense problem

The English past tense is an example of a quasi-regular domain. This problem domain has dual nature – the majority of verbs form their past tense by following a rule for stem suffixation, also referred to as + ed rule. This rule allows for three possible suffixes - /d/ e.g. – tame – tamed; /t/ e.g. – bend – bent and /ed/ - e.g. – talk – talked. However, a significant number of verbs form their past tenses by exceptions to that rule (example: go – went, hide - hid) [4]. The verbs adhering to the former rule based approach are called regular verbs, while the verbs belonging to the second category are called irregular verbs. Also some of the irregular verbs share the characteristics of the regular verbs. For instance, many irregular verbs have regular endings, /d/ or /t/ but with either a reduction of the vowel, example: say – said, do - did or a deletion of the stem consonant, example: has – had, make – made [5]. This overlap between regular and irregular verbs is also a challenge for the model.

Our base model, prior to implementing sources of variation, was inspired by that proposed by Plunkett and Marchman [4], though see [6] for more recent, larger scale models. Plunkett and Marchman suggested that both the regular and the exception verbs could be acquired by an otherwise undifferentiated three-layer backpropagation

network, trained to associate representations of phonological form of each verb stem to a similar representation of its past tense.

3 Methodology

A synergy of ANN and GA is applied to model the system for acquisition of English past tense verbs. The GA component is used to create a population of ANNs and to encode the neuro computational parameters of the ANN into the genome.

The methodology can be summarized as follows:

1. The first step is to design ANNs incorporating a set of computational parameters that would constrain their learning abilities. In our case, we select 8 parameters. These parameters correspond to how the network is built (number of hidden units, architecture), its activation dynamics (the slope of the logistic activation function), how it is maintained (weight decay), how it adapts (learning rate, momentum, learning algorithm), and how it generates behavioral outputs (nearest neighbour threshold).
2. The next step concerns the calibration of the range of variation of each of these parameters. Encoding the parameters within a fixed range allows variation in the genome between members of population, which then produces variations in computational properties. The range of variation of the parameter values serves as the upper and the lower bound used for converting the genotype (encoded values) into its corresponding phenotype (real values).
3. The third step consists of encoding the range of parameter variation in the artificial genome using a binary representation. We are using 10 bits per parameter; overall the genome has 80 bits. The parameters used and their range of variation are given in Table 1.

Table 1: The Genome describing the neuro - computational parameters and their range.

Parameter	Range of variation
No of Hidden Units	6 - 500
Learning Rate	.005 – 0.5
Momentum	0 – 0.75
Unit threshold function (steepness of logistic function)	.0625 - 4
Weight Decay	0.2 – 0.6
Nearest Neighbour Threshold	0.05 – 0.5

The remaining two parameters are the learning algorithm and the architecture, where Backpropagation training and a 3-layer feed forward network are adopted respectively. In our initial implementation, these parameters were not varied. Then, the methodology continues as follows:

4. The fourth step concerns breeding the population of 100 ANNs using this genome.
5. The fifth step focuses on implementing the variation in the quality of environment, accounted for by SES, by means of filtered training sets. An individual's SES is modeled by a number selected at random from the range 0.6-1.0. This gives a probability that any given verb in the full training set would be included in that individual's training set. This filter is applied a single time to create the unique training set for that individual. The range 0.6-1.0 defines the range of variation of SES, and ensures that all individuals are exposed to more than half of the past tense domain.
6. The last step is about training and evaluating training performance and generalisation.

4 The English past tense dataset

The English past tense domain is modeled by an artificial language created to capture many of the important aspects of the English language, while retaining greater experimental control over the similarity structure of the domain [4]. Artificial verbs are monosyllabic and constructed from English phonemes. There are 508 verbs in the dataset. Each verb has three phonemes – initial, middle and final. The phonemes are represented over 19 binary features. The network thus has 3*19 = 57 input units and 3*19 + 5 = 62 units at the output. The extra five units in the output layer are used for representing the affix for regular verbs in binary format.

In the training dataset there are 410 regular and 98 irregular verbs. As this is a radically imbalanced dataset generating a classifier is challenging as the classifier tends to map/label every pattern with the majority class. The mapping of the training set is given a frequency structure, called the token frequency, representing the frequency with which the individual encountered each verb. Some verbs are considered of high frequency whilst others of low frequency. The token frequency is implemented by multiplying the token frequency bit with the weight change generated by the difference between the actual output and the target output. In our experiments, the weight change of high frequency verbs was multiplied by 0.3 and of the low frequency verbs by 0.1.

A second dataset is created to assess the generalisation performance of the model. The main intent is to measure the degree to which a network can reproduce in the output layer properly inflected novel items presented in the input. The generalisation set comprises 508 novel verbs, each of which share at least two phonemes with one of the verbs in the training set, for example *wug – wugged* [7].

5 Experimental settings and performance assessment

A population of 100 ANNs, whose parameters are generated by the GAs, was trained in two different setups. In the first setup, the population of ANN was trained using the full training set, i.e. it contains all the past tense verbs, along with their

accepted past tense forms (henceforth referred as the Non Family setup). In the second setup, we used the filtered training sets, by taking samples from the perfect training set to create subsets, for each member of the population (henceforth referred as the Family setup). This arrangement ensures that each member of the population has a different environment or training set, and thus simulated the effect of SES. Though the networks are trained according to their filtered or Family training sets, the performance is always assessed against the full training set. A comparison of Non Family and Family setups demonstrates the impact of variability in the environment, independent of the learning properties of the ANNs.

We report below results from training 100 feed forward nets, using the batch version of RPROP algorithm. The stopping condition was an error goal of 10^{-5} within 1000 epochs. The performance was assessed using two modes - the MSE with weight decay and the recognition accuracy using nearest neighbours based criteria. The first criterion employed the Hamming distance while the second one was threshold based and used the Root Mean Square (RMS) error.

5.1 Nearest neighbour technique based on Hamming Distance

In the training set, there were 508 monosyllabic verbs, constructed using consonant-vowel templates and the phoneme set of English. Phonemes were represented over 19 binary articulatory features.

The nearest neighbour accuracy was measured between the actual and the target patterns on a phoneme – by – phoneme basis using the Hamming distance. In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In other words, it measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other. This method provides an efficient way of calculating the nearest neighbours. The algorithm for calculating the Hamming distance is listed below.

1. Take the first pattern from the actual output and the desired output.
2. Calculate the Hamming distance between these two patterns, individually, for all three phonemes. This implies that phoneme 1 of actual output is matched with phoneme 1 of the desired output. Similarly, phoneme 2 and phoneme 3 are matched with corresponding phonemes in desired output.
3. **IF** the Hamming distance between all three phonemes is less than 2, then calculate the Hamming distance between last five bits of both patterns.
4. **IF** this distance is equal to zero, then pattern is counted as correct classification, **ELSE** it is counted as an error or misclassification.
5. In the case of misclassification, the last five bits of both the actual output and the desired output are compared with the allomorph (which consists of all possible classes with their binary representations), to find out the actual assigned class and the desired class.
6. **IF** the last five bits of the actual pattern do not match with any pattern of allomorph, then that pattern is classified as ‘random’.

7. In case the **IF** condition specified in **step 3** does not hold, then the same pattern of actual output is matched with the next pattern from the desired output set. This process continues till either the IF condition (of step 3) is satisfied OR till all patterns in the desired output set have been scanned through. In the latter case, if no match is found, then that pattern of actual output set is considered as 'Not Classified'. Repeat the process with next patterns of the actual output set.

This method gives us the total number of correct classifications, total number of errors and the types of errors for each network. The allomorph used in this algorithm is as follows: [0 0 0 0 0] represents Irregular (Ir) verbs; [0 0 1 0 1] denotes a Regular verb with +d rule (R^d); [0 1 1 0 0] stands for a Regular verb with +t rule (R^t); [0 1 0 1 0] denotes a Regular verb with +ed rule (R^ed).

5.2 Nearest neighbour technique based on RMS error threshold

We tested the performance of the networks with an alternative technique, nearest neighbour threshold. The process is as follows:

1. Take the first pattern from the actual output set.
2. For each actual output pattern, starting from the first target pattern, consider all available target patterns.
3. Calculate the root mean square error between the actual and target pattern on a phoneme–by–phoneme basis. This implies, calculating the RMS between phoneme 1 of the actual output pattern and phoneme 1 of target set pattern, and then the RMS values for phoneme 2 and phoneme 3 as well. This results in a 508 row by 508 column array of RMS values, where each array element contains 3 values corresponding to the RMS error between the three phonemes.
4. Based on the range of the nearest neighbour values, as specified in the genome of the artificial neural networks, apply threshold on the RMS values of three phonemes taken together.
5. Select only those neighbours whose RMS error values are lower than the corresponding threshold values.
6. Compare the last five bits of the actual pattern and the selected nearest neighbours with the allomorph to determine their respective classes.
7. Select the 'majority' class from amongst the neighbours and compare it with the class assigned to the actual output pattern. If these matches, then count success else count miss classification. Repeat this process for all the patterns of the actual output set.

6 Results

We report on the classification accuracy and generalisation performance, with respect to three measures – the MSE, the Hamming distance and lastly the nearest neighbour threshold.

In terms of measuring performance based on Hamming distance for all types of verbs, i.e. irregulars, R^d, R^ed and R^t, Table 2 lists the mean values for classification success. Table 3 contains the types of miss classifications the networks made and the mean values of those errors and finally Tables 4 and 5 list the improved results after applying some post processing techniques, discussed below.

Table 2: mean classifications success per category.

Type of verb	Mean Classification on Training Set		Mean Classification on Generalisation Set	
	Non Family Networks	Family Networks	Non Family Networks	Family Networks
R^d	252.24	242.60	265.28	251.69
R^t	71.80	57.56	71.57	57.06
R^ed	10.70	10.89	7.21	7.46
Irreg	5.29	7.22	N.A.	N.A.

Table 3: Types of miss classification errors and their mean values

Assigned Category	Desired Category	Training Set		Generalisation Set	
		Non Family	Family	Non Family	Family
Irreg	R^d	3.68	4.35	3.12	3.97
Random	R^d	8.58	12.05	9.70	12.12
R^d	R^ed	5.06	6.44	8.00	10.08
Random	R^ed	34.70	31.68	41.40	38.93
R^t	R^d	9.15	7.36	9.69	8.00
R^t	R^ed	2.71	1.77	4.39	3.60
Random	R^t	14.47	23.89	17.80	25.99
R^d	R^t	8.81	9.13	7.28	7.66
Irreg	R^ed	4.74	7.61	4.99	6.24
R^d	Irreg	2.80	3.17	0	0
Irreg	R^t	1.26	3.38	1.30	4.54
Random	Irreg	11.69	10.21	0	0
R^ed	Irreg	1.39	1.58	0	0
R^ed	R^t	0.89	0.92	0.85	0.90
R^t	Irreg	1.30	0.87	0	0

Table 3 lists the types of misclassifications made by the population of networks. The most frequent misclassification was classifying a regular verb as a regular but in the wrong category, that is, the incorrect allomorph, e.g. instead of talk – talked (+ed),

networks convert it as talk – talkd (+d or +t). The second most frequent mistake was classifying regular or irregular verbs as *random.* In most cases, this happens due just to the difference of one bit between the actual output affix (last five bits) and the target verb affix.

We do not consider the aforementioned two misclassifications as errors on the following grounds.

- In the first case, it is evident that the network(s) applied the production rule for forming past tense. This implies that the methodology used for converting verb to its past tense is correct.
- In the latter case, the network(s) produces all three phonemes correctly (the phonemes of the actual and target patterns match). The difference of one bit occurs in the last five bits (past tense affix). This indicates that the mechanism followed is correct, especially since the network does not categorise the verb in an incorrect category.

Therefore, we applied post-processing techniques in these two cases, which improved the accuracy of the model.

Table 4: Average performance and improvements on training set

	Non Family Networks		Family Networks	
	Correct in %	Error in %	Correct in %	Error in %
Actual Results	66.9	21.9	72.7	21.6
Improved Results	84.4	4.4	82.8	11.1

Table 5: Average performance and improvements on the generalisation set

	Non Family Networks		Family Networks	
	Correct in %	Error in %	Correct in %	Error in %
Actual Results	67.7	21.3	62.2	24.0
Improved Results	80.0	9.9	75.2	11.1

Improving Performance on training set: We employed three different techniques in order to improve the performance.

1. Considering misclassification amongst regular verbs as okay.
2. Regular verb patterns classified as random due to difference in just 1 bit were considered okay.
3. Irregular verb patterns classified as random due to difference in just 1 bit were considered okay.

Improving Performance on the generalisation set: The following technique was applied to improve generalisation performance.

1. Verb patterns classified as random due to difference in just 1 bit were considered okay.

Our model achieved 84.4 % and 80.0% accuracy on training datasets when used in the Non Family mode and an accuracy of 82.8 % and 75.6 % on generalisation dataset when tested in Non Family and Family modes, respectively. The initial results indicate that classification accuracies are not significantly different in the two modes. For example, in Table 4, the population has the average/mean accuracy of 84.4% when exposed to full training set (Non Family mode) and of 82.8% when exposed to filtered training set (Family mode). This held for generalisation performance as well, as described in Table 5. These results indicate that for the ranges of genetic and environmental variation considered, genetic variation has more influence in determining performance while acquiring past tense.

Performance based on Mean Square Error

As the second measure of performance assessment, the MSE was used to predict the accuracy. The minimum, maximum, mean and the standard deviation of time taken, performance and epochs is reported in Table 6.

Table 6: Performance based on MSE

	Non Family				Family			
	Min	Max	Mean	STD	Min	Max	Mean	STD
Time (seconds)	170	164,653	1,930	387	170	164,668	1,366	976
Performance	0.0502	42.5200	0.1229	0.3765	0.0360	42.4800	0.1100	0.3600
Epochs	425	1000	484	280	542	1000	491	284

Performance based on nearest neighbour threshold

The performance is reported in terms of number of correct classifications and the number of miss classifications made in Table 7.

Table 7: Performance based on nearest neighbour threshold using RMSE

	Non Family				Family			
	Min	Max	Mean	STD	Min	Max	Mean	STD
Correct Classifications	365	414	387	17	364	427	393	25
Misclassifications	94	143	121	17	81	144	115	25

The results show that the average correct classification performance is 76.2% and 77.4% in the discussed modes.

7 Conclusion

In this paper, we explored the use of artificial neural networks as computational models for producing English past tense verbs, and proposed a synergistic approach to capture population variability by (a) combining ANN with genetic algorithms and (b) applying a filter to the training set to simulate environmental influences such as socioeconomic status. The performance of the model was assessed using three different measures and in two different setups. Our tests produced encouraging results as to the utility of this method, and a foundation for future work in using a computational framework to capture population-level variability. These results indicate that for the ranges of genetic and environmental variation considered, genetic variation has more influence in determining performance while acquiring past tense. Our next steps are to consider the impact of different respective ranges of genetic and environmental variation, along with exploring different neural architectures.

References

1. Thomas, M.S.C., McClelland, J.L.: Connectionist models of cognition. In R. Sun (Ed.), Cambridge handbook of computational cognitive modelling (pp. 23-58). Cambridge: Cambridge University Press, (2008).
2. Thomas, M.S.C., Karmiloff-Smith, A.: Connectionist models of development, developmental disorders and individual differences. In R. J. Sternberg, J. Lautrey, & T. Lubart (Eds.), Models of Intelligence: International Perspectives, (p. 133-150). American Psychological Association, (2003).
3. Thomas, M.S.C., Ronald, A., Forrester, N.A.: Modelling socio-economic status effects on language development. Manuscript submitted for publication, (2012).
4. Plunkett, K., Marchman, V.: U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. Cognition, 38. (1991).
5. Lupyan, G., McClelland, J.L.: Did, Made, Had, Said: Capturing quasi – regularity in exceptions. 25th Annual Meeting of the Cognitive Science Society, (2003).
6. Karaminis,T., Thomas, M.S.C.: A cross-linguistic model of the acquisition of inflectional morphology in English and modern Greek. Proceedings of the 32nd Annual Meeting of the Cognitive Science Society, August 11-14, (2010).
7. Thomas, M. S. C., Ronald, A., & Forrester, N. A.: Modelling the mechanisms underlying population variability across development: Simulating genetic and environmental effects on cognition. *DNL Tech report 2009-1*(2009)