Using an evolutionary algorithm to guide problem selection in an on-line educational game.

Elizabeth Sklar Jordan Pollack
Brandeis University
Waltham MA 02454 USA
sklar,pollack@cs.brandeis.edu

Abstract. Traditionally, computer games and educational software define levels within the domains they address and learners track their progress according to the levels they have completed. However, not all domains lend themselves easily to leveling. As well, not all students learn in the same way; following a pre-engineered path through a series of levels may work for some students, but the same path will not work for all students. We present preliminary results from an ongoing experiment in which an evolutionary algorithm is used to guide problem selection for participants in an on-line educational game.

1 Introduction

The Community of Evolving Learners (CEL) system is a framework that enables experimentation with human and machine learning [Sklar,2000]. It is available to anyone with Internet access and a Java-enabled web browser. Inside CEL, participants play educational games with each other and with software agents — and the system adapts as the players learn.

A prototype study was conducted with the CEL system, during the first six months of 1999. Forty-four children from a public primary school (grades 4 and 5) engaged in two typing games. In each game, the children were presented with ten words to type, as fast as they could, maintaining 100% accuracy. Both games were competitive and required two players. Both players received the same list of words to type, and whoever typed each word faster became the "owner" of that word — with the goal of owning the most words at the end of the game.

The two games, called *Keyit* and *Pickey*, are shown in figures 1a and 1b (respectively). They differ only in user interface: in Keyit, words are given one at a time, and players must complete each word before being shown the next in the list; in Pickey, all ten words are displayed at once, and players pick which words they want to type, in any order. The underlying mechanism which supplies word lists to each game is identical. The words are chosen from a standard dictionary of 35000 words.

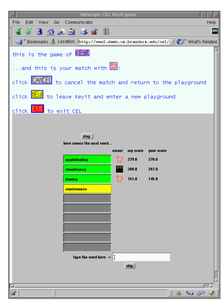


Figure 1a: Keyit.

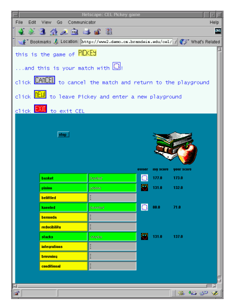


Figure 1b: Pickey.

Every word in the dictionary is characterized according to seven features: (1) word length, (2) keyboarding level[1], (3) Scrabble score[2] (4-7) number of vowels, consonants and 2- and 3-consonant clusters. An evolutionary algorithm [Holland,1975] guides selection of words from this seven-dimensional feature space, adaptively choosing an appropriate *population* of words for each player. Every time a user plays a game, a new *generation* of words is selected, based on the user's past performance with the system[3].

[1] An eleven-level standard was chosen from several that define the order in which keys are presented to students of touch-typing. Each word is assigned the highest keyboarding level of any letter in the word.

[2] Scrabble is a crossword-puzzle style board game, popular in English-speaking countries for many decades.

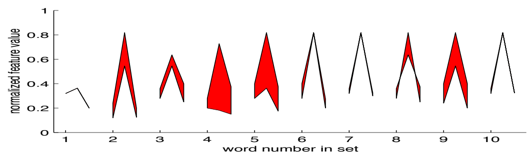
[3] First-time users are given a randomly selected list of words

2 Challenges

In theory, standard reproduction operators, like *crossover* and *mutation*, could be used to produce offspring from a parent generation of feature vectors, each pointing to words in the dictionary. However, the feature space is extremely sparse: over 90 million combinations of the seven feature values exist[4], but only 6074 correspond to words in the dictionary. To combat this problem, we use *reproduction through sampling*. A sample of 1000 words is chosen from the dictionary, and offspring are selected from the sample by first computing a *distance*[5] from each word in the sample to each word in the parent generation. Then the user's performance with each word in the parent generation is compared with his overall average performance. Words with better than average performance are replaced by offspring far away in the feature space (*exploration*); words with worse performance are replaced by offspring that are nearby (*exploitation*).

3 Preliminary results

Some brief and preliminary results are shown in figure 2.



	parent	child	distance
1	altruist (8,4, 8,3,5,3,1)	altruist (8,4, 8,3,5,3,1)	0.000000
2	apt (3,6, 5,1,2,1,0)	repine (6,9, 8,3,3,0,0)	4.714286
3	autocracy (9,7,16,4,5,2,0	perused (7,6,10,3,4,0,0)	6.714286
4	buttock (7,8,15,2,5,2,0)	rides (5,2, 6,2,3,0,0)	18.428571
5	imprinting (10,9,15,3,7,4,1)	stiller (7,4, 7,2,5,2,0)	15.428571
6	inaugurate (10,9,11,6,4,0,0)	snarled (7,9, 8,2,5,2,0)	5.571429
7	menacing (8,9,13,3,5,1,0)	unusually (9,9,12,4,5,2,1)	0.714286
8	nasally (7,9,10,2,5,2,1)	preserved (9,7,15,3,6,2,0)	5.142857
9	purest (6,6, 8,2,4,1,0)	brokenness (10,9,16,3,7,3,0)	14.714286
10	untapped (8,9,13,3,5,2,0)	adventist (9,9,13,3,6,3,0)	0.428571

Figure 2a: Distance between corresponding words from one parent and child generation. The bottom (parent) and top (child) curves for each shape compare the first three (normalized) values in each word's feature vector. The larger a shape's area, the further the child is from its parent in feature space. A mixture of distances should be maintained so that users do not experience an abrupt shift, for example, from easy to hard words.

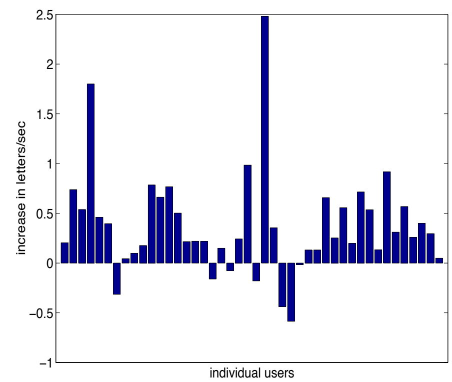


Figure 2b: Change in typing speed. Bars with a positive value show an improvement. The variation in values reflects the fact that some students played many games, while some played only a few; some children were easily distracted and others were not.

4 Discussion

While the typing games described here have limited pedagogical value[6], the children showed genuine enjoyment and interest in the games and the CEL system, especially in being able to interact on-line with their friends while sitting at separate computers. The evolutionary algorithm used to select word lists for each player proved to be a unique and viable method for guiding learners through the domain. Future work involves collaborative projects with education specialists to design CEL activities that adhere to current *constructionist* [Papert,1991] trends, while incorporating some of the techniques described here.

References

Holland, J.H. Adaption in Natural and Artificial Systems, University of Michigan Press, 1975.

Papert, S. Situating Constructionism, *Constructionism*, Ablex Publishing, 1991.

Sklar, E.I. CEL: A Framework for Enabling an Internet Learning Community. Ph.D. Thesis, Brandeis University, 2000 (*expected*).

[4] Considering practical bounds on each feature.

[5] The distance between any two words is the mean squared difference between their feature vectors.

[6] The primary objective was to test the CEL mechanism and the evolutionary algorithm described herein.