A Dual-Route Model that Learns to Pronounce English Words

Roger W. Remington
NASA Ames Research Center
MS 262-2
Moffett Field, CA 94035
rwr@cs.cmu.edu

Craig S. Miller
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3891
cmiller@cs.cmu.edu

Abstract

This paper describes a model that learns to pronounce English words. Learning occurs in two modules: 1) a rule-based module that constructs pronunciations by phonetic analysis of the letter string, and 2) a whole-word module that learns to associate subsets of letters to the pronunciation, without phonetic analysis. In a simulation on a corpus of over 300 words the model produced pronunciation latencies consistent with the effects of word frequency and orthographic regularity observed in human data. Implications of the model for theories of visual word processing and reading instruction are discussed.

Introduction

Mastering English word pronunciation is made difficult by the many inconsistencies in English spelling to sound correspondences. As a result, the skilled reader of English cannot be content with learning a small set of generally applicable rules, but instead must master a large number of highly specific rules and their exceptions. For example, a rule that would pronounce the word 'bough' would have to specify the entire word to distinguish it from 'rough' or 'through.' It is not surprising that many children have great difficulty learning to read English, and many adults remain poor readers. If we understood how knowledge about pronunciation was acquired and represented, it might be possible to design more effective instructional techniques.

Our understanding of how pronunciation knowledge is learned and represented can be furthered by modeling the process with the goal of simulating the behavior of the human learner. This approach has been taken by Coltheart et al (1993), Seidenberg & McClelland (1989), and Seidenberg et al (1995). The model of Coltheart et al (1993) learns symbolic pronunciation rules which specify letter-phoneme correspondences for specific letter contexts. Each rule receives a weight proportional to the number of different words in which the rule applies. This procedure weights letter-phoneme correspondences on the basis of their predictive value across words, rather than on the frequency of occurrence alone. This model generates correct pronunciations for a large proportion of English words.

The models of Seidenberg & McClelland (1989) and Seidenberg et al (1995) learn to pronounce words by adjusting the weights between word features and phoneme units in a connectionist network. Their models are sensitive to the relative frequency of specific letter-phoneme correspondences in specific contexts. Both Seidenberg & McClelland (1989) and Seidenberg et al (1995) train their systems by presenting words in proportion to their (log) frequency of occurrence in English text. Thus, weights in their models are sensitive to both the regularity across words and the frequencies of the individual words in which specific mappings appear. The two connectionist models differ considerably, but each has been successful in accounting for important regularities in isolated word and nonword pronunciation, demonstrating that a uniform process of phonetic analysis can simulate a diverse range of human pronunciation data.

In general, however, the psychological literature suggests that multiple distinct memory systems are involved in learning complex tasks (see Baddeley 1990). Researchers in reading have long debated the role of phonetic analysis and visual recognition in reading individual words (e.g., Paap et al 1987). Differences in learning strategy or ability can result in different patterns of performance. Teaching methods have tended to reflect the changing preferences for phonetic analysis vs. whole-word recognition with little understanding of how this affects reading fluency.

In this paper we report results of simulations that explore how multiple learning algorithms could cooperate to learn to pronounce a large corpus of English words. Because we ultimately simulate the acquisition of reading skill under different instructional techniques, we choose an architecture that can capture the contribution of both phonetic analysis and whole-word techniques. Learning proceeds in two distinct modules: 1) a rule-based module that learns specific letter-to-phoneme rules, and 2) a whole-word module that learns to map word features to the complete pronunciation without phonetic analysis. After training, some words are pronounced by the application of phonetic rules, others by complete or partial mappings of letters to whole word pronunciations.

Model Description

The model is implemented in the Soar cognitive architecture (Newell 1990). In Soar, all knowledge is stored in long-term memory as productions, which fire when their conditions are matched by elements in working memory. When a production fires its output is placed in working memory. In Soar, productions propose, select, and apply operators, which are Soar's basic units for modifying internal representations. When a Soar model initially encounters a problem, it may not yet have the productions needed to propose or select the appropriate operator. At this point, Soar reaches an impasse and must create a subgoal in which other existing knowledge can be used to resolve the impasse. For example, Soar may engage in look-ahead search by trying out one of the com-

peting operators to see if it produces a familiar word. Soar learns by resolving impasses. Once an impasse is resolved, the subgoal knowledge used in the resolution is "chunked" into a single production which will then immediately fire the next time the same problem is encountered. Since a chunk summarizes the performance of several operators, chunking produces more efficient processing resulting in a speed-up in performance. Also, depending on the generality of the chunked production, the selection knowledge may transfer to similar problems. Soar's chunking mechanism is the basis for our model's learning algorithms.

The important functionality of the model is contained in two modules: the rule-based module, and the whole-word module. The rule-based module attempts to construct a pronunciation by phonetic analysis. Knowledge about letter-phoneme correspondences are stored as rules that produce phonemes given a letter context. Prior to training, the rule-based module is given rules for all individual letter-phoneme correspondences (including letter pairs, such as 'th,' that map onto single phonemes). These rules are not sufficient to reliably construct pronunciations. All vowels and some consonants have more than one phoneme correspondence, and the model must learn to resolve this ambiguity by learning rules for choosing the correct phoneme. The greater the consistency in letter-phoneme mapping for a particular context, the greater the opportunity for learning rules that generalize to other words.

The whole-word module begins with no domain knowledge. During training, its algorithm will create rules which map one or more of the letters in a word to the entire pronunciation for the word.

To describe the model's functioning, consider how the model might learn to pronounce its first word, 'dog' [/dog/] (the phonetic notation used in Seidenberg & McClelland, 1989 will be used throughout). The model first attempts to match a pronunciation production to the entire letter string ('dog'). With sufficient exposure to a word there is a high likelihood that a chunk will have been created that associates the entire letter string to the correct pronunciation. If so, the model produces the pronunciation in the minimum number of steps. If no such chunk exists, as is the case upon initial presentation of a word, an impasse results and the model first tries to construct a pronunciation by phonetic analysis in the rule system.

In the rule system, phonetic analysis proceeds by successively selecting a phoneme for each letter working from the beginning of the word to the end. An index pointer is set to the first letter ('d') and rules that assign phoneme values to the letter 'd' are proposed. Since there are some double letters that map onto single phonemes (e.g., 'th' → /T/) and since prior learning may have produced rules which consider the subsequent letter context, rules that map 'do' and 'dog' are also matched. Rules which match a larger context are preferred, thereby creating a bias in favor of rules with more specific context-sensitive assignments during learning. At the outset, however, there are no more specific rules for the 'd' in 'dog' and given that 'd' is unambiguous, the phoneme /d/ is chosen as the first phoneme, and the index pointer updated to the 'o'.

The vowel 'o' has several phonetic realizations. An impasse is reached because there is no knowledge yet to use to select one of the options. A subgoal is created to resolve this impasse. In the current implementation, operators in the subgoal choose randomly from the set of possible phonemes and then attempt to complete the word pronunciation. If the correct pronunciation is generated, the model learns by associating the correct pronunciation of the 'o' with the subsequent two letters. For the 'o' in 'dog' it will build a rule that prefers the phoneme /o/ for the letter 'o' when followed by a word-final 'g'.

If the model chooses correctly in this subgoal, it returns the correct phoneme for 'o' to the phonetic analysis subgoal. If the model chooses incorrectly, our current implementation expends no further resources and simply guesses among the remaining operators. In either case, the index pointer advances to 'g', which can either be realized as a hard or soft 'g'. Again, an impasse results and the subgoal process is repeated for 'g'.

If the rule-based module correctly assembles the pronunciation, it summarizes its processing by building a chunk that produces /dog/ from 'dog'. It may also have built a rule to pronounce 'o' when followed by word-final 'g' and/or a rule to pronounce word-final 'g'. The chances of correctly pronouncing 'dog' on the first try are about 1 in 12. The initial set of letter-phoneme rules specify 6 alternate phonemes for 'o' and 2 for 'g'. Both mappings must be correctly assigned before learning will occur. Likewise, the rule for 'o' followed by word-final 'g' has a 1 in 12 chance of being learned, since again both mappings must be correctly assigned. The final 'g' rule however has a 1 in 2 chance of being learned. In general, our model is biased to resolve ambiguities at or near the end of a word before those at or near the beginning.

Repeated exposures to the word 'dog' will increase the likelihood of one or more of the rules being learned. Likewise, repeated exposure to words with word-final 'g' and with word-final 'og' will eventually produce rules that resolve those ambiguities. Thus, with a few exposures to 'dog' and 'log' there is a reasonable probability of quickly generating pronunciations for 'bog', 'cog' and even nonwords such as 'mog'. The model will learn less useful chunks if initially exposed to many words with irregular letter-phoneme correspondences. Since the more regular letter-phoneme correspondences occur more often by definition, there is a greater probability of learning regular correspondences in the rule-based module.

Initially, the chance of correctly pronouncing a word is small, even for a very regular word like 'dog'. If the phonetic analysis fails the whole-word module attempts to generate a pronunciation. The algorithm used in the whole-word module is adapted from the Symbolic Category Acquisition (SCA) algorithm of Miller (1993). Input to SCA is an object defined by features; output is a category to which the object is assigned based on the set of features. Here, SCA treats the letters in a word as features, and the pronunciation as the category. As training progresses SCA builds productions that map an increasing number of letters in a word to the pronunciation. In this way, repeated exposure leads to more specific productions, until at the end there is a production that associates the entire letter string with the correct pronunciation. It is possible to learn to pronounce a word solely by the whole-word module. Once there is a chunk that matches the entire

letter string that chunk will fire in the first stage of processing and will be pronounced in minimum time, regardless of whether that chunk was created in the rule-based module or the whole-word module.

Because letter strings are always being matched to whole pronunciations, the whole-word system does not learn productions of general utility. In contrast, productions in the rule-based system specify single letter pronunciations for specific contexts that can appear in many words. This distinction captures an important difference between whole-word reading and phonetic analysis.

Simulation

We evaluated the model by training it on a corpus of words of varying frequency and orthographic regularity. The combined effects of word frequency and orthographic regularity produce a highly reliable pattern of pronunciation times in human data. High frequency words are pronounced more rapidly than low frequency words. Orthographically regular low frequency words are pronounced more rapidly than orthographically irregular low frequency words. But, regularity has no effect on high frequency words, many of which are irregular. If our model correctly simulates pronunciation difficulty, then it should produce pronunciation latencies that at least preserve the ordinal relationships seen in the human data.

We use the number of Soar operators selected as a measure of pronunciation latency. The greater the number of operators required for a task, the greater the difficulty and, hence, the greater the latency. Our model requires a minimum of 2 operators for a fully learned word: one to perceive the letter string, and one to generate the pronunciation. If a word is not fully learned, additional operators will be needed for the phonetic analysis and the whole-word generation. The maximum number of operators depends on the nature of the input and prior exposure and cannot be calculated directly. The maximum observed in this simulation was 46 operators.

J. McClelland graciously provided us with the word list used by Seidenberg & McClelland (1989). We created a word list containing regular and exception words by selecting items from the Seidenberg & McClelland word list that had already been categorized on the basis of orthographic regularity (see Seidenberg & McClelland, 1989). Consistent and regular words have regular orthography and were both categorized as regular. Exception words and some strange words (e.g. 'aisle') have irregular orthography and were grouped together as exception words. The strange words with consistent, regular orthography (e.g. 'yelp') were categorized as regular. This yielded a total of 208 regular words and 127 exception words.

To simulate word frequency, the number of exposures to each word was determined by dividing its log frequency by the log frequency of the least frequent word. This procedure maintains the ratio of log frequencies between words. For analysis, the resulting frequency ratios were then partitioned into three equally spaced frequency categories. Each word was thus categorized as a 'high' 'medium' or 'low' frequency word. The resulting list was randomized to avoid biasing the model by systematic presentation effects. Five repetitions of the list were run and statistics computed after each run.

Table 1: Model Latencies by Frequency

Freq	Regular	Exception	E-R
Low	10.12	12.20	2.08
Med	3.50	4.22	.72
High	2.27	2.34	.07

Table 2: Model Latencies by Training Cycle

Freq	Run	Regular	Exception	E-R
Low	1	20.04	21.67	1.63
	2	11.50	14.15	2.65
	3	7.71	10.28	2.51
	4	6.13	8.26	2.13
	5	5.26	6.64	1.38
Med	1	8.86	12.63	3.77
	2	2.67	2.50	-.17
	3	2.00	2.00	0
	4	2.00	2.00	0
	5	2.00	2.00	0
High	1	3.33	3.70	.43
	2	2.00	2.00	0
	3	2.00	2.00	0
	4	2.00	2.00	0
	5	2.00	2.00	0

Table 1 shows a frequency by regularity interaction similar to that seen in the human data. The advantage of regularity seen for low frequency words is systematically reduced with increasing frequency. Table 2 shows the data broken down by runs. The advantage of regularity decreases in all frequency groups as the number of runs increases. However, this decrease is much less for the lowest frequency words. Interestingly, with enough exposure all words are pronounced very quickly. This too is a feature of the human data. Good readers do not show the effect of orthographic regularity for low frequency words, presumably because they have seen even moderately low frequency words many times. Like our model, human readers seem to be sensitive to the absolute amount of exposure, not just to the relative frequency.

The rule-based system learns chunks that resolve letter ambiguities by looking at the immediate letter context. These rules should generalize to words with similar contexts, speeding up the learning of new words. This trend appears in Table 2. If the rule-based system is learning useful chunks, generalization should be more effective with regular words than exception words. Table 3 shows that 59% of the correct recognitions occurred in the rule-based module, 41% in the whole-word module. The conditional probability of a rule-based solution given a regular word was p(rule | regular) = .66, while p(whole | regular) = .34. The rule-based system pronounced almost twice as many regular words as the whole-word module. In contrast, the conditional probabilities for exception words was: p(rule | exception) = .46, p(whole | exception) = .54. The whole-word module pronounced a slightly greater proportion of exception words than the rule-based module. This is consistent with our initial expectation that analysis should be more effective for regular words than for irregular words.

Table 3: Relative Frequency of Process Method

	Rule-Based	Whole-Word
Regular	.44	.23
Exception	.15	.18

Discussion

We have shown how even a simple two-process model can account for important aspects of human data on visual word pronunciation.

One obvious concern is that we have produced only qualitative fits, relying on the overall similarity of patterns between the human response time results and operator count, which measures the computational complexity of the problem for our Soar model. A related concern is that the model learns too quickly. Performance is at asymptote after only 15 exposures to a corpus of over 300 words. Surely, this exceeds human learning rates. Consider, however, that there are many parameters that could be adjusted to improve the correspondence to response time and learning rate data. Changing the input features from letters to letter features, for example, would alter the learning rate of both modules, create chunks that mapped fragments of two letters onto a phoneme, and affect the contexts over which generalization would occur. Likewise, by tinkering with the phonetic analysis and the SCA algorithm we could fine tune the number of operators. For example, gradually increasing the specificity of the chunks with training would alter both the learning rate and the relative number of operators for regular and exception words. And so on. For present purposes though we deliberately avoided the temptation to adjust parameters to achieve a better fit. Because we are striving for a model with breadth that could also simulate different methods of reading instruction, it seemed prudent to explore a very simple, straightforward architecture, and not risk overfitting by arbitrarily adjusting parameters.

A reliance on ordinal fits also avoids assumptions about the relative scale properties of response time and operators. Newell (1990) derives a estimate of 60-120 msec for a decision cycle which has proven useful in fitting data in some contexts. Each decision cycle represents the selection of an operator, and any such estimation assumes that each operator takes an approximately constant time. This approximation may fail because it does not adequately reflect brain processing, or because operators within a model are not matched for computational complexity. Ordinal fits make fewer potentially erroneous assumptions.

What do we feel are the theoretically important features of our model? Certainly, there is a theoretical stance taken in using a dual-route approach. Because we wish ultimately to model the effects of different instructional methods, it is important to explore the hypothesis that they produce different internal representations. The model also asserts that phonetic analysis rules are taught, not inferred from practice. The effect of practice is to condition the application of the rules. The model currently has no way of creating phonetic rules without explicit knowledge of the individual letter-phoneme correspondences. Without this knowledge all learning will be done in the whole-word system, whose rules will not generalize. This is a strong assertion. Later implementations may relax this to enable the model to reason about the possible letter-phoneme relationships in words it has learned in the whole-word module. However, we know of no data that would suggest that such reasoning is done implicitly when reading, nor data that would suggest that the rules for phonetic analysis can be learned implicitly from whole-word instruction/reading. Our current hypothesis is that if children can learn phonetic rules by inference, then it is not a by-product of reading, but a separate deliberate process.

Another feature of the model is the assumption that fully learned words are "recognized" and not pronounced by phonetic analysis. Phonetic analysis occurs only for words that have not been fully learned. With enough exposure then, many words will be pronounced as whole-words. Currently, this exposure is in terms of absolute frequency. The higher a word's frequency, the more often it is encountered, and the greater the opportunity for one of the two system to learn it completely. The only effect of relative frequency is to alter the probability that a given word will be encountered. It is not clear yet whether the strong form of this is correct. The evidence that good readers show no difference in pronunciation times for regular and irregular words suggests that the absolute number of times a word is encountered is important. Our model perhaps exaggerates the effect of absolute frequency by learning so quickly, but this serves the useful purpose of focusing interest on this factor.

Finally, the model is sensitive to the order in which it encounters training examples. If given regular words, it will learn chunks that can be usefully generalized. If first exposed to irregular words, the chunks will be less useful. Again, the effect may be exaggerated by the simplicity of the model, but this too leads to interesting and testable predictions for reading instruction.

A simple dual process model of human visual word pronunciation was presented that successfully simulates the combined effects of word frequency and orthographic regularity. The simplicity of the model exaggerates the effects of certain factors, such as absolute frequency and order effects in training, providing useful insights into factors which may also affect how we learn to read.

Acknowledgements

This research was sponsored in part by the McDonnell Foundation, Grant JSMF 91-34, and by the Markle Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the McDonnell Foundation or the Markle Foundation.

References

Baddeley, A. (1990). *Human Memory: Theory and Practice.* Boston, MA: Allyn and Bacon.

Coltheart, M., Curtis, B., Atkins, P., & Haller, M. (1993). Models of reading aloud: Dual-route and parallel distributed processing. *Psychological Review, 100,* 589–608.

Miller, C. S. (1993). *Modeling Concept Acquisition in the Context of a Unified Theory of Cognition.* PhD thesis, The University of Michigan. Also available as Technical Report CSE-TR-157-93.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Paap, K. R., McDonald, J. E., Schvaneveldt, R. W., & Noel, R. W. (1987). Frequency and pronounceability in visually presented naming and lexical decision tasks. In Coltheart, M. (Ed.), *Attention and Performance XII: The Psychology of Reading*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Seidenberg, M. S. & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96*, 523–568.

Seidenberg, M. S., Plaut, D. C., Peterson, A. S., McClelland, J. L., & McRae, K. (1994). Nonword pronunciation and models of word recognition. *Journal of Experimental Psychology: Human Perception and Performance, 20*, 1177–1196.