Pronunciation by Analogy and Naming Latency

Y. Marchand
Institute for Biodiagnostics (Atlantic)
1796 Summer Street, Suite 3900
Halifax, Nova Scotia
Canada B3H 3A7

R. I. Damper
School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK

Abstract

In the long-standing debate about the cognitive mechanisms of reading aloud, Glushko's latency data play a crucial role. The debate has often been characterized in terms of dual-route versus single-route theories. In this paper, we study the capability of a single-route computational model based upon pronunciation by analogy (PbA) to account for Glushko's main findings from his three experiments. Candidate features to explain these effects in the PbA framework are processing time and various measures of complexity of the pronunciation lattice that is used for generating the pronunciation of a spelling pattern. While none of these is able by itself to replicate all the results of the three experiments, the processing time is a good predictor of the wordness effect and the number of lattice arcs is a good predictor of the regularity effect.

Introduction

Pronunciation by analogy (PbA) is a technique for the automatic phonemization of text that we have developed extensively for use in text-to-speech synthesis, strongly influenced by the early work of Glushko (1979) on the cognitive process of reading aloud. In seeking to understand how human readers convert print to sound, intense debate has raged over whether there is a single (lexical) route or dual (lexical plus rule-based) routes. Dual-route theories (Forster and Chambers 1973; Coltheart 1978; Coltheart, Rastle, Perry, Langdon, and Ziegler 2001) assume that words are pronounced by lexical look-up whereas non-word letter strings can only be pronounced by abstract letter-to-sound rules. Glushko carried out a series of three important experiments that "refute current claims that words are read aloud by retrieving a single pronunciation from memory and that pseudowords are pronounced by using abstract spelling-to-sound rules". In his experiments, subjects read aloud monosyllabic words and pseudowords (i.e., nonwords of English, which nonetheless conform to the spelling patterns of the language) that resembled them. His naming latency data constitute strong evidence against dual-route theories.

Data from dyslexic readers have figured prominently in arguments for dual routes. Two of the most important dyslexias are *phonological* and *surface* dyslexia. Patients with phonological dyslexia display good ability to read words (both regular and irregularly spelled) aloud but poor nonword reading ability (Beauvois and Dérouesné 1979). In surface dyslexia, however, patients misread irregularly spelled words, which tend to be regularized in their pronunciation (Coltheart, Masterson, Byng, Pryor, and Riddoch 1983). The usual interpretation is that the presumed dual routes to reading are differentially affected. That is, in phonological dyslexia the rule-based route is impaired while the lexicon is relatively spared but in surface dyslexia the rules are intact while the lexical route is disrupted. Until recently, PbA has not been seriously considered as a model of impaired reading, although we have shown (Damper and Marchand 2000) that, in spite of being a single-route model, it is capable of reproducing the neurological phenomena of phonological and surface dyslexias by selectively impairing its component parts. Here, we extend this previous work to naming latency.

Pronunciation by Analogy

In PbA, an unknown word is pronounced by matching substrings of the input to substrings of known, lexical words, hypothesizing a partial pronunciation for each matched substring from the phonological knowledge, and assembling the partial pronunciations. Here, we use an extended and improved version of the PRONOUNCE system described by Dedina and Nusbaum (1991), which consists of four components: the lexical database, the matcher that compares the target input to all the words in the database, the pronunciation lattice (a data structure representing possible pronunciations), and the decision function that selects the 'best' pronunciation among the set of possible ones (Figure 1).

Lexicon

The lexicon used by PRONOUNCE is *Webster's Pocket Dictionary*, containing approximately 20,000 words. For PbA, it is necessary to *align* each letter of a word with a corresponding phoneme in strict one-to-one fashion, as briefly explained below. Dedina and Nusbaum's version of the dictionary was aligned by a "simple Lisp program that only uses knowledge about which letters and phonemes are consonants and which are vowels" (p. 58). Unfortunately, further details of this simple program are unavailable, but it is unlikely that it makes a very good job of the difficult task of alignment.

A manually-aligned version of this dictionary, used by Sejnowski and Rosenberg (1987) for training their NETtalk neural network for text-to-phoneme conversion, is publicly available via the World Wide Web from sub-directory `dictionaries` at `ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/`. To cater for those cases where a word's spelling has more letters than its pronunciation has phonemes, yet still allow a strict one-to-one correspondence, Sejnowski and Rosenberg's introduced a null phoneme symbol. To allow for the (rarer) cases where a word's spelling has fewer letters than its pronunciation has phonemes, they

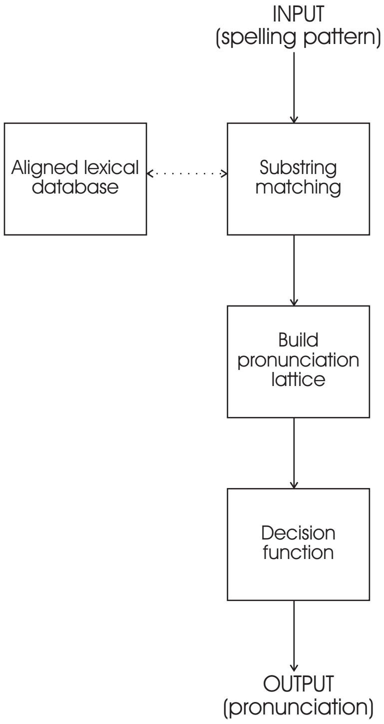

Figure 1: Component parts of Dedina and Nusbaum's PRONOUNCE.

introduced 'new' phonemes, such as /X/ to represent the spelling-to-sound correspondence <x> → /ks/ in words like <sex>. This is the version of *Webster's Pocket Dictionary* used in this work, except that for consistency with our previous work, homonyms (413 entries) were removed from the original NETtalk dataset to leave 19,596 entries.

Pattern Matching

An input word is matched in turn against all orthographic entries in the lexicon. For a given entry, the process in PRONOUNCE starts with the input string and the dictionary entry left-aligned. Substrings sharing contiguous, common letters in matching positions are then found and information about these matching substrings and their corresponding aligned phoneme substrings is entered into a pronunciation lattice as described below. One of the two strings is then shifted right by one letter and the process repeated, until the input string and the dictionary entry are right-aligned. (We call this 'partial' pattern matching.) It follows that PbA requires a dictionary in which text and phonemes have been previously aligned, so that pronunciations corresponding to matching orthographic substrings can be identified.

Pronunciation Lattice

The pronunciation lattice is a graph data structure used to represent all the matching substrings (and, thereby, the possible pronunciations) of an input word. A node of the lattice represents a matched letter, L_i, at some position, i, in the input. The node is labeled with its position index i and with the phoneme that corresponds to L_i in the matched substring, P_{im} say, for the mth matched substring. An arc is placed from node i to node j if there is a matched substring starting with L_i and ending with L_j. The arc is labeled with the phonemes intermediate between P_{im} and P_{jm} in the phoneme part of the matched substring. Additionally, arcs are labeled with a 'frequency' count (note that this has nothing to do with the frequency of occurrence of a word or subword string in the language) that is incremented each time that substring, with that pronunciation, is matched during the pass through the lexicon. Figure 2 shows an example lattice for the pseudoword ('nonword') <shead>. For clarity of presentation, this lattice has been simplified by excluding many nodes and arcs.

As an illustration of how to interpret the pronunciation lattice, consider the arc between the *Start* node and the node labeled /ɛ/ at position 3. This arc is labeled with the pronunciation /ʃ_/ (where /_/ represents the null phoneme) and with a frequency count of 10. This means that there were 10 words in the dictionary that matched the pseudoword <shead> from the start of the word up to the letter <e> at position 3 *and* had the pronunciation /ʃ_ɛ.../. These 10 words were: <shed>, <shelf>, <shell>, <shellfish>, <shelter>, <shelve>, <shelves>, <shepherd>, <sherrif> and <sherry>.

Decision Function

A possible pronunciation for the input corresponds to a complete path through its lattice, from *Start* to *End* nodes, with the output string assembled by concatenating the phoneme labels on the nodes/arcs in the order that they are traversed. (Different paths can, of course, correspond to the same pronunciation.) Scoring of candidate pronunciation uses two heuristics:

1. If there is a unique shortest path, then the corresponding pronunciation is taken as the output.

2. If there are tied shortest paths, then the pronunciation corresponding to the best scoring of these is taken as the output.

The choice of scoring function(s), so as to determine the 'best' candidate pronunciation, is a major consideration in the design and implementation of any PbA model.

Recent Improvements

The implementation used here features several enhancements over Dedina and Nusbaum's original PRONOUNCE (Marchand and Damper 2000). First, we use 'full' pattern matching between input letter string and dictionary entries, as opposed to the 'partial' matching of PRONOUNCE. That is, rather than starting with the two strings left-aligned, we start with the initial letter of the input string $\mathcal{I}$ aligned with the last letter of the dictionary entry $\mathcal{W}$. The matching process terminates not when the two strings are right-aligned, but when the last letter of $\mathcal{I}$ aligns with initial letter of $\mathcal{W}$. Second, multiple (five) heuristics are used to score the candidate pronunciations. (Although the focus of this paper is naming latency, and we will not have a great deal to say about the actual pronunciations produced by our PbA system, or how they were produced, these details are included for completeness.) Individual scores are then multiplied together to produce

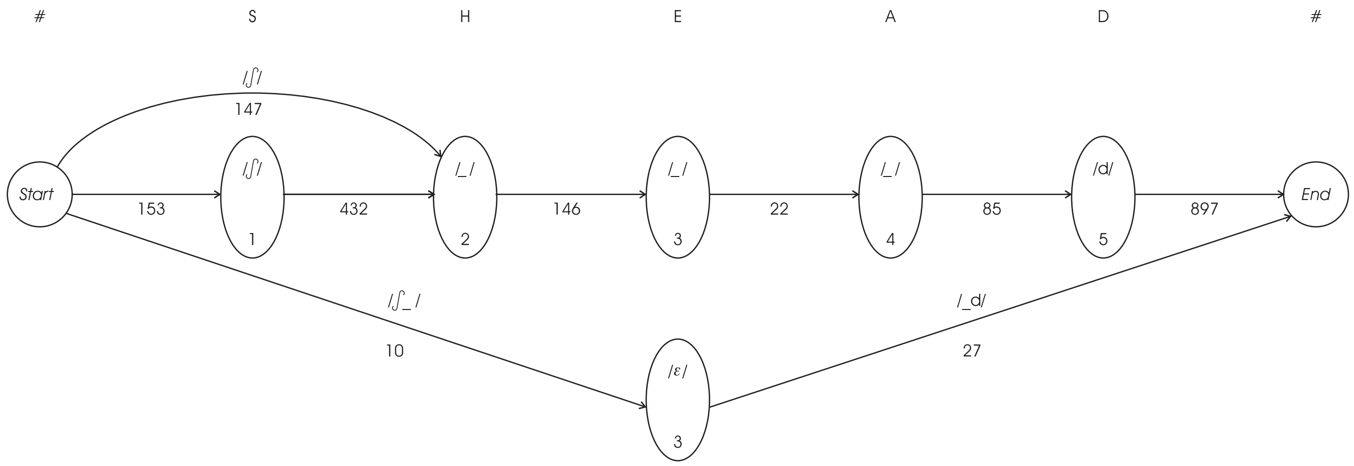

Figure 2: Example pronunciation lattice for the pseudoword <shead>. The delimiter symbol for the space at the beginning and end of the word is #. For simplicity, only a subset of nodes and arcs is shown.

a final overall score and the best-scoring pronunciation is selected as output. Marchand and Damper show that this 'multistrategy' approach gives statistically significant performance improvements over simpler versions of PbA.

Review of Glushko's Experimental Results

Experiment 1: Glushko examined the differences in pronunciation and latencies between four kinds of spelling patterns generated from 43 monosyllabic *exception* words, such as <deaf>. The four kinds of stimuli were: (a) regular words, (b) exception words, (c) regular pseudowords, (d) exception pseudowords. He wrote: "Exception words are words with different spelling-to-sound correspondences than most words with the same vowel and terminal consonants". As far as possible, Glushko chose a matching *regular* word that differed from the base exception word only in its terminal consonant. Regular and exception pseudowords were then generated from the regular and exception words by changing the initial consonants.

As in previous studies, Glushko found that pseudowords took longer to read on average than words and that exception words, which "break the rules", took longer to read than regular words. His most important new result, however, was that exception pseudowords took longer to pronounce than regular pseudowords. That is, the latency for pseudowords is affected by the existence of orthographically-similar words having irregular spelling-to-sound correspondences, in contradiction of the original formulation of dual-route theory. A two-way analysis of variance with main factors *wordness* and *regularity* showed the significant naming latency advantage a stimulus enjoys by being a word and by being regular.

Pronunciation errors were much lower for regular letter strings than for exception letter strings. For 17.6% of the exception pseudowords, subjects pronounced the string with an "irregular" correspondence found in an exception word that it resembled. This should not happen if there are entirely separate lexical and orthographic routes to reading, with the pronunciation of pseudowords generated solely by the latter mechanism. Rather, there is a lexical influence on nonword pronunciation. Of the errors on exception words, 84% of these were regularizations. Again, this should not happen if the pronunciation of words is independently generated by retrieval from lexical memory.

Experiment 2: Glushko anticipated two possible objections to his interpretation of the results of Experiment 1, namely: (1) some "priming" took place leading to an influence of word pronunciation on pseudoword pronunciation; (2) since words and pseudowords were randomly mixed in the stimuli, subjects may have adopted their strategy for word recognition and pronunciation, and adapted this to the pronunciation of nonwords. Experiment 2 was designed to overcome these two possible artifacts, which were eliminated by including only pseudowords in the stimulus list.

There were 16 subjects in Experiment 2. Exception pseudowords (26 in all) were formed by changing the initial consonant or consonant cluster of an exception word. In all cases, the initial consonant was a stop to reduce the variability in pronunciation latency. Matching regular pseudowords were then constructed by changing the terminal consonant, ensuring that this gave a regular vowel-coda rhyme. Contrary to the prediction based on the assumed existence of abstract rules, the latency advantage of regular pseudowords over exception pseudowords in Experiment 1 was maintained.

Experiment 3: In Experiments 1 and 2, Glushko made a distinction between regular and exception words that effectively redefined the traditional distinction. He introduced a classification that "… may be more appropriate for psychology" of *consistent* and *inconsistent* words. Words are deemed consistent or inconsistent according to the orthographic and phonological structure that they activate during the analogical process, i.e., according to the consistency of pronunciation of their orthographic neighbours—whether or not they rhyme. With this new definition, Glushko tested a three-way classification of words into: exception, regular and consistent, regular and inconsistent.

There were 16 subjects in Experiment 3. Each of 41 exception words like <have> was paired with a regular and (necessarily) inconsistent word like <wave>. Then each exception and each regular/inconsistent word was (as far as possible) matched with a regular/consistent word that differed only in the terminal consonant. For analysis, the 82 consistent control words were collapsed irrespective of whether they were derived from an exception word ("exception control") or a regular word ("regular control"). Results showed that regular/consistent words were pronounced more rapidly than either regular/inconsistent words or exception words, as predicted by the analogical hypothesis. Further, the pronunciation error rate was significantly lower for the consistent words (0.5%) than for the regular/inconsistent (2.9%) or exception words (8.3%). For the exception words, 84% of the errors were regularizations. For the regular/inconsistent words, 53% of the errors were irregularizations.

PbA Simulations of Naming Latency

Here, we report on simulations of Glushko's experimental data using the PbA model. Various ways were used to simulate the naming latency using PbA. First, we simply measured the computer processing time required to produce a pronunciation in our simulations. Although important, this direct time measure does not give us much insight into why naming latency differences appear between different categories of spelling patterns. Hence, we also examined some indices of the complexity of the pronunciation lattice as possible predictors of latency, namely the numbers of nodes, arcs, pattern matchings and competing, candidate pronunciations. The approach was to test these features of the pronunciation lattice for their ability to replicate Glushko's data about naming latencies. All the following statistics assume two-tailed significance tests.

Experiment 1: Table 1(a) shows the processing time plus the complexity measures of the pronunciation lattice obtained using the stimuli of Experiment 1. All these measures were subjected to two-way analysis of variance (ANOVA) in which the factors were *wordness* (words versus nonwords) and *regularity* (regular versus exception). Table 1(b) shows the results of this statistical analysis. This direct measure of naming latency reproduced one important aspect of Experiment 1, namely the wordness effect. PbA generated faster the pronunciation of words (mean 112.8 ms) than the pronunciation of pseudowords (mean 119.4 ms) on a 733 MHz OptiPlex GX110 workstation. Moreover, when all the four categories of stimuli were collapsed ($N = 172$), the Pearson correlations between the processing time and the number of nodes, arcs and pattern matchings were respectively 0.634, 0.730 and 0.836 ($p < 0.01$ in all three cases). These high correlations are perhaps not surprising since the pronunciation lattice is the core data structure used to represent the output of the matching process within PbA.

Table 1(b) indicates that the the number of nodes is a poor predictor of the wordness and regularity effects found in Glushko's Experiment 1, failing to reach significance in both cases. This is not at all surprising, since the number of nodes in a letter string depends fairly directly upon the number of letters and the number of phonemes that these individual letters map to. But the Glushko stimuli do not differ systematically in their length (4 or 5 letters long) and attempts were made in their construction to control the letters used. Hence, as Table 1(a) shows, there is little difference in the number of nodes between the four categories of stimuli.

On the other hand, according to Table 1(b), the number of arcs seems to be an excellent predictor of the wordness effect, and a reasonable predictor of the regularity effect. From Table 1(a), regular words and pseudowords are seen to have fewer arcs on average (76.4) than exception words and pseudowords (81.6). For the wordness effect, however, the differences are in the wrong direction! The words own more arcs (mean 83.4) than the pseudowords (mean 74.6) despite the fact that words are read faster. Thus, naming latency can not be directly proportional to the number of arcs in the pronunciation lattice.

Considering next the number of candidates, on average the pronunciation lattice of a pseudoword yields significantly more candidate pronunciations (mean 3.5) than the lattice of a word. In the case of a word, there is always a complete match of the input to a word in the lexicon, so that the number of candidates is identically 1 (with a shortest path length of 1)—hence the standard deviation of 0 in Table 1(a) that leads to the significant wordness effect seen in Table 1(b). Some advantage is also seen for regular spelling patterns (mean 3.07) versus exception ones (mean 3.5) but this does not reach significance. So, for Experiment 1, the best predictors of the wordness effect are the processing time and the number of candidates. The best predictor of the regularity effect is the number of arcs, but this is only rather weak. Unfortunately, no one measure manages to predict both wordness and regularity effects together.

Experiment 2: Experiment 2 was designed to eliminate two possible artifacts in Experiment 1, namely priming of non-word pronunciations by the activation of word pronunciations and biasing the subjects' strategy in the direction of pronouncing words. This he did by excluding words from the list of stimuli. Of course, neither of these cognitive effects is yet modeled in PbA, so our simulations are insensitive to them. Accordingly, we feel free to interpret our results without considering such effects.

Table 2(a) shows the processing time plus the complexity measures of the pronunciation lattice obtained using the stimuli of Experiment 2. Table 2(b) shows the results of the paired-sample t-tests that were conducted on the features of the pronunciation lattices between regular and exception pseudowords. As before, the processing time replicates the significant advantage ($p < 0.05$) a regular letter string (word in this case) enjoys over its irregular counterpart. Further, the number of arcs is also a significantly good predictor ($p < 0.05$) of the regularity effect. Taken together with the marginally significant result of the Experiment 1 simulation, this strengthens the case for number of arcs as an effective measure of regularity. Finally, the number of candidates is a marginally significant predictor of the regularity effect in this simulation ($p < 0.1$).

Experiment 3: Table 3(a) shows the processing time plus the complexity measures of the pronunciation lattice obtained using the stimuli of Experiment 3. Because Experiment 3 used only words, the number of candidates is identically equal to 1 (because there is only one shortest path in the case of a word listed in the dictionary) so this feature is not considered. All other features were subjected to a paired-sample t-test. This test was realised between: (i) exception words and regular consistent words; (ii) regular inconsistent and regular consistent words; (iii) all inconsistent words

Table 1: PbA simulation results for Glushko's Experiment 1. (a) Means and standard deviations (in brackets) for the features of the pronunciation lattices. (b) ANOVA analysis. KEY—* : $p < 0.1$; ** : $p < 0.05$; *** : $p < 0.01$; NS: Not significant.

	Words		Pseudowords	
	Regular	Exception	Regular	Exception
Processing time (ms)	113.3 (23.9)	112.3 (24.0)	119.3 (27.6)	119.5 (27.4)
Nodes	26.02 (4.49)	26.35 (4.43)	25.74 (4.39)	26.14 (5.00)
Arcs	80.72 (22.06)	86.16 (22.73)	72.16 (18.06)	77.00 (23.65)
Pattern matchings	6108 (2534)	5875 (2475)	6021 (2419)	5819 (2688)
Candidates	1.00 (0.0)	1.00 (0.0)	5.14 (2.77)	6.00 (4.08)

(a)

	Processing time		Nodes		Arcs		Matchings		Candidates	
Effect	*F*	Sig.	*F*	Sig.	*F*	Sig.	*F*	Sig.	*F*	Sig.
Wordness	6.23	**	0.35	NS	18.34	***	0.12	NS	118.49	***
Regularity	0.01	NS	0.55	NS	3.02	*	0.58	NS	1.73	NS
Interaction	0.12	NS	0.15	NS	0.22	NS	0.53	NS	1.73	NS

(b)

Table 2: PbA simulation results for Glushko's Experiment 2. (a) Means and standard deviations (in brackets) for the features of the pronunciation lattices. (b) Paired-sample t-tests between regular pseudowords and exception pseudowords with $df = 25$. KEY: as for Table 1.

	Regular pseudowords	Exception pseudowords
Processing time (ms)	114 (26.09)	124 (24.81)
Nodes	25.96 (4.69)	26.04 (4.64)
Arcs	72.69 (22.8)	79.04 (26.6)
Pattern matchings	6022 (2515)	6366 (2685)
Candidates	3.96 (3.16)	5.31 (2.95)

(a)

Processing time		Nodes		Arcs		Matchings		Candidates	
t	Sig.	t	Sig.	t	Sig.	t	Sig.	t	Sig.
−2.63	**	−0.17	NS	−2.11	**	−0.89	NS	−1.72	*

(b)

Table 3: PbA simulation results for Glushko's Experiment 3. (a) Means and standard deviations (in brackets) for the features of the pronunciation lattices. (b) Paired-sample t-tests between various categories of stimuli: (i) exception versus regular consistent words; (ii) regular inconsistent versus regular consistent words; (iii) all inconsistent versus consistent words. KEY: as for Table 1.

	Exception ($N = 41$)	Regular inconsistent ($N = 41$)	Regular consistent ($N = 82$)
Processing time	120.5 (25.88)	119.3 (24.94)	117.7 (25.50)
Nodes	26.50 (4.81)	26.88 (5.15)	26.09 (4.69)
Arcs	87.61 (24.77)	85.73 (23.64)	80.15 (22.49)
Pattern matchings	6225 (2719)	6300 (2922)	6288 (2733)

(a)

	Processing time		Nodes		Arcs		Matchings	
	t	Sig.	t	Sig.	t	Sig.	t	Sig.
(i)	0.82	NS	0.82	NS	1.98	*	−0.03	NS
(ii)	0.50	NS	0.96	NS	1.21	NS	−0.13	NS
(iii)	−0.93	NS	−1.27	NS	−2.27	**	0.11	NS

(b)

(exception plus regular) and consistent words.

Results are shown in Table 3(b). Recall that Glushko's main finding for Experiment 3 was a latency advantage for regular consistent words over inconsistent words—either regular inconsistent or exception, with the latter taking slightly longer to pronounce that the former. This pattern of values is seen in the processing times (Table 3(a)) but the differences are not statistically significant. The only feature yielding any significant differences is the number of arcs.

Discussion and Conclusions

A single-route model of reading aloud based upon PbA reproduces quite well important results on naming latencies. Unfortunately, no single measure of pronunciation 'complexity' is able to explain the whole range of experimental findings of Glushko (1979). However, processing time is a good predictor of the wordness effect seen in Experiment 1, whereby words are read faster than pseudowords. Further, one measure of the complexity of the pronunciation lattice, namely the number of arcs, proved to be a generally excellent predictor of the regularity effect, whereby regular and/or consistent words are read faster than exception and/or inconsistent words. While the number of arcs also gave a highly-significant result for the wordness effect in Experiment 1, this was in the 'wrong' direction.

References

Beauvois, M. F. and J. Dérouesné (1979). Phonological alexia: Three dissociations. *Journal of Neurology, Neurosurgery and Psychiatry 42*(12), 1115–1124.

Coltheart, M. (1978). Lexical access in simple reading tasks. In G. Underwood (Ed.), *Strategies of Information Processing*, pp. 151–216. New York: Academic Press.

Coltheart, M., J. Masterson, S. Byng, M. Pryor, and J. Riddoch (1983). Surface dyslexia. *Quarterly Journal of Experimental Psychology 35A*, 469–495.

Coltheart, M., K. Rastle, C. Perry, R. Langdon, and J. Ziegler (2001). DRC: A dual-route cascaded model of visual word recognition and reading aloud. *Psychological Review 108*(1), 204–256.

Damper, R. I. and Y. Marchand (2000). Pronunciation by analogy in normal and impaired readers. In *4th Conference on Natural Language Learning and 2nd Learning Language in Logic Workshop*, Lisbon, Portugal, pp. 13–18.

Dedina, M. J. and H. C. Nusbaum (1991). PRONOUNCE: A program for pronunciation by analogy. *Computer Speech and Language 5*(1), 55–64.

Forster, K. I. and S. M. Chambers (1973). Lexical access and naming time. *Journal of Verbal Learning and Verbal Behavior 12*, 627–635.

Glushko, R. J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance 5*(4), 674–691.

Marchand, Y. and R. I. Damper (2000). A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics 26*(2), 195–219.

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