Paper for publication in *Language and Speech*, final version, April 20, 1997.

Pronunciation by Analogy: Impact of Implementational Choices on Performance

R.I. Damper* and J.F.G. Eastmond

*Cognitive Sciences Centre
and
Image, Speech and Intelligent Systems (ISIS) Research Group,
Department of Electronics and Computer Science.

University of Southampton,
Southampton SO17 1BJ,
UK.

Center for Spoken Language Understanding,
Oregon Graduate Institute of Science and Technology,
PO Box 91000,
Portland,
OR 97291-1000,
USA.

(Emails: `{rid|je}@ecs.soton.ac.uk`)

Short running head: Pronunciation by Analogy

This work was supported by research grant R000235487 "Speech Synthesis by Analogy" from the UK Economic and Social Research Council.

Address for correspondence:

Until 1 July 1997:

Dr. R.I. Damper,

Image, Speech and Intelligent Systems (ISIS) Research Group,

Department of Electronics and Computer Science,

University of Southampton,

Southampton SO17 1BJ,

UK.

Tel: +44 1703 594577

FAX: +44 1703 594498

Email: `rid@ecs.soton.ac.uk`

After 1 July 1997:

R.I. Damper, PhD,

Visiting Research Professor,

Center for Spoken Language Understanding,

c/o Department of Computer Science and Engineering,

Oregon Graduate Institute of Science and Technology,

PO Box 91000,

Portland,

OR 97291-1000,

USA.

Tel: +1 503 690-1517

FAX: +1 503 690-1306

Email: `damper@cse.ogi.edu` or

`rid@ecs.soton.ac.uk` (will auto-forward)

ABSTRACT

Pronunciation by analogy (PbA) is an emerging, data-driven technique with potential application in text-to-speech (TTS) systems, as well as being an influential psychological model of reading aloud. The underlying idea is that a pronunciation for an unknown word (i.e. one not in the dictionary, or lexicon, of the human or machine 'reader') is assembled by matching substrings of the input to substrings of known, lexical words, hypothesising a partial pronunciation for each matched substring from the lexical knowledge of the 'reader', and concatenating the partial pronunciations. This paper assesses the capability of PbA to derive pronunciations for unknown words of English.

As a psychological model, PbA is 'underspecified', i.e. the implementor of a simulation of the process faces detailed choices which can only be resolved by trial and error. One goal for this paper is to explore the impact of certain basic implementational choices on the performance of PbA systems. The variables studied are the specific lexical database used as the basis of the analogy process, the way of ranking/scoring candidate pronunciations, and the effect of manual versus automatic alignment of letters and phonemes.

When tested with short (monosyllabic) pseudowords previously used in experimental psychology studies, the lowest error rate achieved is 14.3% (for a test set of size 70). We conclude that current PbA systems are at best poor models of pseudoword pronunciation by humans.

To assess their suitability for use in a TTS application, in which multisyllabic words will be encountered, the implementations have also been tested with lexical words temporarily removed from the dictionary. The best performance obtained was 93.5% phonemes correct (corresponding to 67.9% words correct) for a 16,280-word dictionary. This is vastly superior to the 25.7% words correct obtained using a set of popular letter-to-sound rules, indicating considerable scope for analogy methods to be exploited in future TTS systems.

INTRODUCTION

There is an obvious commonality between the computational problem of text-to-speech (TTS) conversion in speech synthesis and the psychological process of reading aloud (Sullivan & Damper, 1989). Given this, it is perhaps surprising that there has been so little synergy between the two fields of speech technology and experimental psychology. From one point of view, psychological evidence and theories can give fresh insights into the kind of ‘hard’ computational mapping problems encountered in TTS conversion, as well as offering promise for more human-like output from the machine. From the other point of view, the success (or otherwise) of a computational implementation represents an acid test of sufficiency for any psychological model of language processing.

Against this background, there seems great potential for cross-fertilisation between the related endeavours of TTS conversion and computational modelling of reading aloud by humans. One particular area where such cross-fertilisation does seem to have occurred in a modest way is that of pronunciation by analogy (PbA). In a TTS system based on analogy, most words would be pronounced by retrieving their phonemic ‘baseform’ (Lucassen and Mercer, 1984) from the machine’s dictionary, or lexicon. The pronunciation for a ‘novel’ word absent from the lexicon, however, is derived not by the application of abstract[1] letter-to-sound rules but is ‘assembled’ from the (known) pronunciations of words that it resembles. As an approach to synthesis, such methods often take their inspiration from experimental psychology, where pronunciation by analogy is an influential model of reading aloud.

A small number of computational implementations of PbA has been described in the literature. Dedina and Nusbaum (1991) outline a system for English, called PRONOUNCE, which was actually implemented some years before (Dedina & Nusbaum, 1986) with the purpose of testing the computational sufficiency of pronunciation by analogy. In a series of papers, Sullivan and Damper (1990; 1991; 1992*a*; 1992*b*; 1993) describe a rather more complex and developed system for English and German. However, PbA systems turn out to have important similarities to TTS systems which are founded on the paradigm of case-based reasoning (e.g. Lehnert, 1987; Stanfill, 1987; 1988; Golding, 1991; Golding & Rosenbloom, 1991), which take their inspiration from artificial intelligence (AI) rather than from the psychology of reading.

As a psychological theory, pronunciation by analogy is seriously under-specified (see below) to the extent that it fails to offer any meaningful guidance on the host of implementation choices which confront anyone programming a computer simulation. While Dedina and Nusbaum give little or no consideration to this point, Sullivan and Damper (1993) show that detailed implementational choices can have a significant impact on performance, and this impact is different for different languages. In this paper, we seek to gain further insight into implementational choices in PbA for English as a way of assessing its ultimate suitability – both as a model of the human process of reading aloud, and as a component of a text-to-speech system.

The remainder of this paper is structured as follows. In the next section, we review the background to this work as defined by psychological models of pronunciation by analogy. Then, existing computational PbA systems are detailed (including those featuring case-based reasoning). Next, we describe the specific PbA system studied (actually a re-implementation of PRONOUNCE) and, in the following section, we describe the lexical databases used as the basis of the analogy process. Results obtained with the 'standard' Dedina and Nusbaum model as well as with somewhat improved versions of it are then presented. The next section compares the performance of PbA with that of a particular set of letter-to-sound rules for deriving the pronunciation of unknown lexical words. Finally, we conclude with some discussion of the issues raised in this study, and some indications for future work.

PSYCHOLOGICAL BACKGROUND TO PbA

Systems for text-to-speech conversion and computational models of the psychological process of reading aloud have been treated rather separately in the past, with different backgrounds and traditions. Sejnowski and Rosenberg's (1987) connectionist system NETtalk represents probably the main point of departure from this generalisation: it is difficult to classify NETtalk unequivocally as either one or the other and, indeed, these authors present it as both.

For each of these traditionally-separate processes, there exists a standard approach or model. In the case of reading aloud, the standard model is *dual-route theory* (e.g. Coltheart, 1978), according to which there is a lexical route for the pronunciation of known words and a parallel route utilising abstract letter-to-sound rules for the pronunciation of unknown, or novel, words (Fig. 1). In the case of letter-to-sound conversion within a TTS system, the standard approach (e.g. Klatt, 1987, Fig. 30, p. 768) also has two routes. It utilises a pronouncing dictionary for known words and a set of general-purpose, context-dependent translation rules (e.g. Ainsworth, 1973; Elovitz *et al*, 1976; Hunnicutt, 1976) which is invoked if the input word is not in the system's dictionary. Unlike the essentially parallel psychological model, dictionary matching and rule application work in sequence in a TTS system.

**** FIGURE 1 ABOUT HERE ****

Arguments for dual-route theory are based on the ability to pronounce pseudowords (non-words which nonetheless conform to the spelling patterns of English), differences in the times taken ('latencies') to read regular and exception words, and apparent double dissociation between the two routes in dyslexia (see Humphreys and Evett, 1985, for an extensive review and discussion). However, it has been variously argued that all these observations can be explained by a single route. One pervasive idea in the literature is that pseudowords are pronounced by analogy with lexical words that they resemble (Baron, 1977; Brooks, 1977; Glushko, 1979; 1981; Kay & Marcel, 1981; Brown & Besner, 1987). Glushko, for instance, showed that "exception pseudowords" like *tave* take longer to read than "regular pseudowords" such as *taze*. Here, *taze* is considered as a "regular pseudoword" since all its orthographic 'neighbours' (*raze, gaze, maze* etc.) have the regular vowel pronunciation /eɪ/. (Note that the phoneme symbols used in this paper are in accordance with Sullivan, 1988.) By contrast, *tave* is considered to be an "exception pseudoword" since it has the exception word (*have*, /hav/) as an orthographic neighbour. Thus, in the words of Glushko (1979), the "… assignment of phonology to non-words is open to lexical influence" – a finding which is at variance with the notion of two separate, independent routes to pronunciation. Instead of this:

> "… it appears that words and pseudowords are pronounced using similar kinds of orthographic and phonological knowledge: the pronunciation of words that share orthographic features with them, and specific spelling-to-sound rules for multiletter spelling patterns."

Thus, in place of *abstract* letter-to-sound rules in the dual-route model we have *specific* patterns of correspondence in the single-route, analogy model.

There are two forms of pronunciation by analogy: *explicit* and *implicit*. Explicit analogy (e.g. Baron, 1977) assumes that the lexicon has a localist, or individuated, representation of words (e.g. in the form of a 'list'): the strategy is then to recall a similar word and modify its pronunciation. In implicit analogy (e.g. Brooks, 1977), a pronunciation is derived from generalised phonographic knowledge about existing words, most usually thought of as represented in a distributed fashion. Implicit analogy has obvious commonalities with most single-route, connectionist models (e.g. Sejnowski & Rosenberg, 1987; Seidenberg & McClelland, 1989) in which the generalised knowledge is learned (for example, by error back-propagation) and stored as a set of network connection weights, and the network has no holistic notion of the concept 'word'. This commonality is pointed up by Glushko's arguments (1979, pp. 686–687; 1981, pp. 71–72) for the term *activation* in place of *analogy* since, for him, the process in humans is naturally implicit. (See Andrews, 1989, for additional discussion.)

Until the recent advent of computational simulations of the process, analogy 'theory' could only be considered seriously underspecified (c.f. Coltheart *et al*, 1993). Clearly, its exact operation must depend critically on some measure of similarity, and "… without a metric for similarity and without a specification of how similar is similar enough, the concept of analogy by similarity offers little insight" (Glushko, 1981, p. 72). Pollatsek and Rayner (1993, p. 416) state: "Current 'analogical' models are a promising attempt to model the access of sound, but … they are currently largely 'hand waving': few of the details have been worked out". As pointed out earlier by Brown and Besner (1987), the operation of lexical analogy must be constrained by factors such as:

- the size of the segment shared between novel and lexical word;
- its position in the two strings;
- its frequency of occurrence in the language;
- and the frequency of occurrence of the words containing it;

none of which details had received serious consideration in previous work but are clearly central concerns for a computational simulation. Accordingly, they write:

> "Extant analogy models are not capable of predicting the outcome of assembly operations for all possible strings."

In particular, the 'theory' gives no principled way of deciding the orthographic neighbours of a novel word which are deemed to influence its pronunciation. What, for instance, are the orthographic neighbours of *joov* – a string which subjects have no problem pronouncing as /ʤuv/? Take the pseudoword *pook*. Do *look* and *pool* have equal status as neighbours? Glushko (1981, p. 73) argues for the primacy of the terminal vowel-consonant – or *body* in Patterson and Morton's (1985) modified dual-route model – in reading, so favouring *look* as a neighbour. The argument is on the basis of latency effects in pronunciation.

While problems such as these can be addressed by implementation of a computational model which can be tested for sufficiency and falsifiability (the approach we strongly favour), Brown and Besner chose to do something different. They experimented by asking their subjects to read aloud pseudowords "for which pronunciation options are limited and relevant neighbours specifiable." For instance, they used *-cc-* pseudowords, such as *haccid*, *haccede* and *accid*, since *-cc-* is most commonly pronounced /k/ in English (e.g *account*), and less frequently as /ks/ (e.g *accident*), with the specific pronunciation almost entirely predictable from the following vowel. Remarkably, however, the majority pronunciation was /s/, yet *cc* → /s/ is very rare indeed (*flaccid* is the sole exemplar.) Brown and Besner's proposition is that "the system charged with generating oral reading uses all the information it can get". Thus, they argue, two parallel strategies are employed in parsing the *-cc-* input: the first reduces *cc* to *c* on the grounds that medial double consonants are never phonemic in English while the second strategy retains *cc*. Both parsings are then applied to a 'flexible' phonological rule system to yield a candidate set, selection from which is influenced by lexical analogy. According to this view, "it is *phonological* similarity which counts".

Briefly, this then forms the psychological background to pronunciation by analogy. What is the computational background?

EXISTING PbA SYSTEMS

As described in the Introduction, existing implementations of PbA take their inspiration either from the psychology of reading, or from the AI paradigm of case-based reasoning. In spite of this dissimilarity, the two kinds of system turn out to have much in common.

PbA Models Based on the Psychology of Reading

We describe here two such systems in a little detail: Dedina and Nusbaum's (1991) and Sullivan and Damper's (e.g. Sullivan & Damper, 1993). The former, PRONOUNCE, was the basis of the present study, while the latter (which is somewhat more developed) is described to provide additional understanding of implementational choices. PRONOUNCE is an explicit system in our terms, since the dictionary is retained in its entirety, whereas Sullivan and Damper's system is implicit in that lexical knowledge is pre-compiled into a knowledge base and the dictionary then discarded. We regard both types of system as 'data-driven', on the grounds that pronunciations are inferred solely and entirely from the lexical database. Recent PbA systems have also been described by Federici, Pirrelli and Yvon (1995) and Yvon (1996).

Dedina and Nusbaum's PRONOUNCE: this was designed to explore the sufficiency of pronunciation by analogy as a psychological theory by testing its underlying assumptions in a computational simulation. According to these authors (p. 57): "pronunciation by analogy may provide the same pronunciation ability as a set of spelling-to-sound rules without requiring an explicit theory of rule induction … and may be relatively simple to automate." They identify the principal theoretical issue as "the degree to which orthographic consistency in the spelling patterns of words is related to phonotactic consistency in pronouncing these words."

Dedina and Nusbaum describe PRONOUNCE [2] as consisting of four basic components: the lexical database, the matcher which compares the target input to all the words in the database (hence explicit analogy), the pronunciation lattice (a data structure representing possible pronunciations), and the decision function, which selects the 'best' pronunciation from among the set of possible ones. However, it is also necessary to align the letters and phonemes of the entries in the lexical database, so that the partial pronunciations for contiguous matched letter substrings can be assembled into a full pronunciation for the complete input string. Hence, the overall structure is as shown in Fig. 2.

**** FIGURE 2 ABOUT HERE ****

The lexical database of PRONOUNCE consists of "approximately 20,000 words based on *Webster's Pocket Dictionary*" in which text and phonemics have been automatically aligned. Dedina and Nusbaum acknowledge the crude nature of their alignment procedure, saying it "was carried out by a simple Lisp program that only uses knowledge about which phonemes are consonants and which are vowels."

The system is tested with novel strings, i.e. pseudowords. An incoming pseudoword is matched in turn against all orthographic entries in the lexicon: there is no 'training' (in the sense of pre-compilation of lexical knowledge) as such. For a given dictionary entry, the process starts with the input string and the dictionary entry left-aligned. The number of contiguous, common letters in matching positions in the two strings is then found, and information about these matching letter substrings – and their corresponding phoneme substrings in the dictionary entry under consideration (hence the need for aligned text and phonemes) – is entered into the input string's pronunciation lattice as detailed below. The shorter of the two strings is then shifted right by one letter and the process repeated. This continues until the two are right-aligned, i.e. the number of right shifts is equal to the difference in length between the two strings.

Matched substrings, together with their corresponding phonemic mappings as found in the lexicon, are used to build the pronunciation lattice for the input string. A node of the lattice represents a matched letter, L_i, at some position, i, in the input. The node is labelled with its position index i and with the phoneme which 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 labelled with the phonemes intermediate between P_{im} and P_{jm} in the phoneme part of the matched substring. Note that arcs corresponding to bigrams are labelled with the empty string: the two symbols of the bigram label the nodes at either end. Additionally, arcs are labelled with a "frequency" count (Dedina and Nusbaum's term, and nothing to do with frequency of usage in written or spoken communication) which is incremented by one each time that substring (with that pronunciation) is matched during the pass through the lexicon. Finally, there is a *Start* node at position 0 and an *End* node at position $(n + 1)$, for an input string of length n – so that word-start and word-end can be considered as special orthographic symbols. Example lattices are given in the results section below.

A possible pronunciation for the input string then corresponds to a complete path through its lattice from *Start* to *End*, 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.) PRONOUNCE traverses the lattice (by unspecified means) to find the set of candidate pronunciations which are then passed to the decision function. Two heuristics are used at this stage to rank the pronunciations; the top-ranking candidate is then selected as the output[3]. The first heuristic is based on path length. If one candidate pronunciation corresponds to a unique shortest path (in terms of number of arcs, each representing a matched segment) through the lattice, this is selected as the output. Otherwise, those candidates that tie in terms of shortest path length are ranked on the sum of their arc "frequencies", reflecting the number of matched substrings that produced that arc.

From this description, it is seen that the status of orthographic 'neighbour' is determined implicitly, primarily by the lengths of matching substrings (irrespective of where they appear) in the input string and in the lexical entries.

Dedina and Nusbaum tested PRONOUNCE on 70 of the pseudowords employed by Glushko in his 1979 study. All these words "were four or five characters long and were derived from monosyllabic words by changing one letter". Seven human subjects with phonetics training were asked to read these and give a transcription for the first pronunciation which came to mind. A 'correct' pronunciation for a given pseudoword was considered to be one produced by any of the subjects in response to that string. PRONOUNCE is reported to have exhibited a word error rate of 9%. (By contrast, DECtalk is said by Dedina and Nusbaum to have an error rate of 3% on these 70 pseudowords.) The authors interpret their results as showing (p. 62):

> "... that pronunciation-by-analogy is computationally sufficient to generate reasonable pronunciations for short novel strings ... *[and]* may, with sufficient development effort, replace the use of spelling-to-sound rules in the next generation of text-to-speech systems."

Sullivan and Damper's System: this system differs from Dedina and Nusbaum's in several ways. Sullivan and Damper (1990; 1991; 1992*a*; 1992*b*; 1993) employ an improved automatic alignment procedure based on the Lawrence and Kaye (1986) algorithm. By pre-computing mappings and their statistics for use in the matching process, they have implemented a considerably more 'implicit' form of pronunciation by analogy: there is no explicit matching of the input string with lexical entries. This pre-computation from the lexicon of possible mappings and their statistics amounts to a form of training. A similar pronunciation lattice is used, except that the nodes represent junctures[4] between symbols rather than the symbols – letters or phonemes – themselves, and arcs represent letter-phoneme mappings. This latter form of lattice can be interpreted as the state diagram of a weighted finite-state transducer and means that all the results and formalisms of that paradigm (Mohri, forthcoming) can be exploited within Sullivan and Damper's system. Hence, we believe that this form of lattice may well be superior (see below).

Sullivan and Damper have also examined several different ways of numerically ranking the candidate pronunciations, taking into account issues such as the probabilities of the letter-phoneme mappings used in the assembled pronunciation. Following Brown and Besner (1987), the analogy process is extended to the phonemic – in addition to the orthographic – domain. This latter extension has necessitated a reversion to some form of rules – “flexible” letter-to-sound rules in Brown and Besner’s terminology – in order that ‘plausible’ pronunciations for an unknown word can be generated and matched against the pre-compiled lexical knowledge. Sullivan and Damper take “flexible” to mean ‘context-independent’, and use Lawrence and Kaye’s (1986) tabulated correspondences, with context dependencies ignored, as their rules. Since the intention is to produce a *set* of plausible pronunciations, there is no necessity to resolve conflicts to ensure that only a single rule is fired (as there would be in a conventional, rule-based TTS system).

Given the improved alignment and candidate-ranking methods, we note in passing that improved performance relative to Dedina and Nusbaum might be expected. However, Sullivan and Damper’s best result for orthographic analogy on the full set of 131 pseudowords from Glushko (1979) (plus another 5 words – see below) is only 70.6% (1993, p. 449). This is an error rate of almost 30%, as compared to Dedina and Nusbaum’s 9% error rate on the rather smaller test set of size 70. Differences in test-set size seem insufficient to explain a discrepancy of this magnitude, as do differences between British and American English and the lexicons employed. (Sullivan and Damper’s best result was obtained using the 800 words of Ogden’s (1937) *Basic English*.) We return briefly to this discrepancy later.

It is worth noting that Sullivan and Damper actually achieved better performance (78.7%) for *phonemic* analogy and using a rather different way of scoring candidate pronunciations.

Case-Based Reasoning

There has been some interest in applying the AI paradigm of *case-based reasoning* (or memory-based reasoning – MBR) to the design of prototype TTS systems. In effect, these are a form of PbA. We describe here the MBRtalk system of Stanfill (1987): similarly motivated systems have been described by, for example, Lehnert (1987).

In Stanfill's words, the basic principle of MBR is that "best-match recall from memory" can be regarded as a "primary inference mechanism". In MBRtalk, for every letter of every word in the dictionary of aligned orthographic-phonemic word pairs, a *frame* is created having 5 fields: the letter itself, the previous 4 letters, the succeeding 4 letters, the (single) phoneme aligned to that letter, and the stress assigned to it. The complete set of frames is stored in memory for use during transcription. In the case of a (possibly novel) input word, there will be one frame for each letter of the word but the *phoneme* and *stress* fields of these frames will be empty; it is the task of MBRtalk to fill them. This is done by comparing each frame of the test word with every frame in memory and retrieving the best-matching frame. The relevant contents of the retrieved memory frames are then transferred to the test frames to give a pronunciation.

This strategy of finding the best match to a lexically-specified pronunciation stored in memory makes it clear that MBRtalk is using a form of (implicit) pronunciation by analogy (see also Damper, 1995). The frames are effectively primitive matched (fixed, 9-character) substrings which provide a single phoneme in the output pronunciation (corresponding to the single, central letter). The simple mechanism of stepping the 9-character window through the input word a character at a time and concatenating output phonemes corresponding to the central letter avoids any need for a pronunciation lattice as an explicit data structure. With a 1024-frame (approximately 200-word) test set, Stanfill found a best accuracy of 88% frames correct with a 132,072-frame (about 25,000 word) training dictionary. In subsequent work, Stanfill (1988) presents a rule-induction methodology within the framework of the MBR paradigm.

More recently, van den Bosch and Daelemans (1993) have described a very similar approach to memory-based reasoning that they describe as "a link between straightforward lexical lookup and similarity-based reasoning". The method takes a pronouncing dictionary as the training set but "solves the problem of lacking generalisation power and efficiency by *compressing* it into a text-to-speech *lookup table*". For Dutch, they say: "The most surprising result of our research is that the simplest method (based on tables and defaults) yields the best generalisation results, suggesting that previous knowledge-based approaches were overkill".

RE-IMPLEMENTATION OF PRONOUNCE

Because of its simplicity, and its status as an early and well-known PbA system, Dedina and Nusbaum's PRONOUNCE was chosen to provide the basis of the current work. PRONOUNCE was re-implemented with a view to assessing its ability to derive pronunciations for unknown words. A second goal was to assess the impact of implementational choices on performance. As well as testing on short pseudowords, we wished also to assess how PbA would perform on 'real' words, including longer ones, as these are arguably more like the novel input strings that a TTS system might encounter.

Critical Appraisal of PRONOUNCE

Although an excellent candidate for a 'baseline' or benchmark against which to assess PbA, PRONOUNCE does appear to have some shortcomings which it is as well to make explicit.

First, the described alignment algorithm seems problematic (see pp. 71–73 of Sullivan, 1992, for discussion) because of its extreme simplicity, based only on the vowels/consonant distinction. In this work, therefore, we have additionally used data which have been manually aligned (not by us). Our expectation is that manual alignment will produce a generally better result than automatic alignment, mainly because human expert knowledge is brought to bear. (Indeed, most automatic alignment algorithms will fail for some inputs.) However, there is one way in which the automatic alignment might be superior. Manual alignment is typically (if not always) done on a strict one-to-one basis, with a single letter aligned to a single phoneme and *vice versa*. When the letter and phoneme forms of a word have different lengths, null or additional 'phonemes' are created to maintain the one-to-one nature of the mapping. Not only is the invention of a new 'phonemes' a clumsy and *ad hoc* device, the one-to-one mapping also fails to recognise that letters often operate together as 'functional spelling units' (e.g. Coltheart, 1984, p. 69) corresponding to a single phoneme. Thus, we are relying on the analogy process itself to effect this grouping into functional units. On the other hand, the strict one-to-one mapping enforced by manual alignment removes any possibility (provided we count null and special symbols appropriately) of length errors in the output – a factor which ought if anything to improve performance.

There are also difficulties with substring matching. The fact that matching starts with the two strings (input and lexical entry) left-aligned and ceases when they are right-aligned is to some degree well-motivated, as it prevents prefixes being matched to suffixes and *vice versa*. We show below, however, that it leads to other problems. The form of the pronunciation lattice, where nodes correspond to the 'ends' of matched substrings (rather than to the junctures between symbols) is also problematic. As a consequence, pronunciations have to be assembled from both arc and node labels. Again, we illustrate below that this can lead to otherwise-avoidable pronunciation failures. The scoring procedure, based on two prioritised heuristics, is *ad hoc*. Apart from the fact that gives primacy to exact, lexical matches (which could as well be treated as special cases), there seems to be no especially good reason to favour shorter pronunciation paths (i.e. longer substring matches) absolutely over more common substring matches, and the summing of arc frequencies in the decision function is intuitively less satisfactory than taking products – since the latter is closer to a probabilistic formulation.

Finally, there is some question about test materials. PRONOUNCE was intended primarily to test the sufficiency of pronunciation by analogy as a psychological theory – so that it was natural to test its performance on pseudowords. However, we have emphasised throughout that analogy-based systems are also of interest as possible components of text-to-speech systems. In this context, ‘real’ words (e.g. lexical words temporarily removed from the dictionary) form a more appropriate test set than short pseudowords, not least because they are multi-syllabic. Temporary removal from the lexicon means that the pronunciation must be assembled by the analogy process rather than merely retrieved in its entirety.

Re-Implementation Details

The re-implementation was programmed in C on a Hewlett-Packard 712/80 workstation running HP-UX. Automatic alignment (where used), substring matching and building the graph (pronunciation lattice) are all done as closely to Dedina and Nusbaum’s implementation of PRONOUNCE as their description allows.

A ‘direct’ re-implementation scores candidates using Dedina and Nusbaum’s method with its two prioritised heuristics (shortest path length followed by maximum sum). We call this model D&N. Two other methods for scoring have also been implemented. Both are based on products of arc ‘frequencies’ which, as noted above, come closer to a probabilistic formulation than do summed frequencies. Hence, our first alternative replaces the maximum sum heuristic with the maximum product of the arc frequencies: we call this model PROD. It still selects primarily on the basis of shortest path length. Our second alternative uses a single heuristic. First, the arc-frequency product along each possible path from *Start* to *End* is taken. Then, for all paths corresponding to the *same* pronunciation, these values are summed to give an overall score for that pronunciation. We call this the TP (total product) model. Of the three, it comes closest to estimating a probability for the assembled pronunciation.

Dedina and Nusbaum do not say precisely how they searched the graph for the ‘best’ pronunciation. There are many ways that this might be done, with varying efficiency. In this work, we have used a simple and direct implementation of depth-first search as a pre-order tree traversal. For the D&N and PROD models, partial paths were pruned during the traversal when their length exceeded the shortest complete path found so far for that input, to reduce run times. A similarly motivated, beam-search pruning was carried out for the TP model: if any product fell below a threshold during traversal, its corresponding path was discarded. The threshold used was ϵ times the maximum product score found so far, with ϵ set by trial and error at 10^{-3}. While this may have led to the pruning of a path contributing to the 'best' pronunciation, the contribution would be very small. Again, this led to a very significant improvement in run times for the testing of lexical words (see below) but it was unnecessary for the testing of pseudowords because of the much smaller size of the test sets in this case.

LEXICAL DATABASES

To examine any impact that the specific lexical database might have on performance, we have used two in this work: the 20,009 words (spellings and pronunciations) of *Webster's Pocket Dictionary* (1974 edition) and the 16,280 words of the *Teacher's Word Book* (TWB) (Thorndike and Lorge, 1944). In both cases, versions exist in which letters and phonemes have previously been manually aligned for the purposes of training back-propagation networks. While our purposes are different, the same aligned data suffice.

The Webster's database has been used in two forms: (initially) unaligned and manually aligned. The former was the version used by Dedina and Nusbaum. Before its use in this work, letters and phonemes were aligned using our re-implementation of their alignment algorithm. The latter manually-aligned version was prepared by Sejnowski and Rosenberg (1987) for training NETtalk[5]. According to our version, there are than 20,012 as stated by Sejnowski and Rosenberg. The dialect of the Webster's database is unequivocally general American English.

The *Teacher's Word Book* database is that used by McCulloch, Bedworth and Bridle (1987) to train NETspeak. Although originally American English, and supplied to us as such, our version of the machine-readable database appears in places to have been manually anglicised. (It has entries such as *colour*.)

The phoneme inventory (at least, as used by us) is of size 52 in both cases, including the null phoneme (for the manually-aligned data) but excluding stress symbols. We leave the (clearly important) problem of stress assignment for later study.

RESULTS OF PbA

In this section, we compare results for the three models (D&N, PROD and TP) using both pseudowords and lexical words as input. We also compare performance using the two lexical databases (Webster's and TWB) for both automatically- and manually-aligned text and phonemes.

Pseudowords

Pronunciations have been obtained for:

- the 70 pseudowords from Glushko (1979) used by Dedina and Nusbaum to test PRONOUNCE. The 'correct' pronunciation for these strings is taken to be that given by Dedina and Nusbaum (1991, pp. 61–62) as produced by their 7 experimental subjects. We refer to this test set as D&N 70. A minor difficulty with this test set is the authors' use of both /o/ and /oʊ/ in their transcriptions without further explanation. (In fact, their transcription standard is undefined.) We show below, however, that this had no effect on results.

- the full set of 131 pseudowords from Glushko plus two other pseudowords (*goot*, *pome*), two lexical words (*cat* and *play*), and the pseudohomophone *kwik*, as used by

Sullivan (1992). The 'correct' pronunciations are those transcribed by Sullivan from the British Received Pronunciations of his 20 non-phonetician subjects. We refer to this test set as Sull 136. Our expectation is that the error rate will be relatively high for this test set, because the subjects' dialect of English is inconsistent with the American lexical databases.

The output has been scored on words correct and also on symbol score (i.e. phonemes correct) using the Levenshtein string-edit distance (Levenshtein, 1966) as shown in Table I.

*** TABLE I ABOUT HERE ***

Our best comparison with Dedina and Nusbaum (D&N 70 test set, D&N model, Webster's database, automatic alignment) gives a figure of 75.7% words correct. This is considerably poorer than their report of approximately 91% words correct, although as far as we are able, we have reproduced their implementation, reference pronunciations, and test set. This result accords more closely with Sullivan and Damper (1993) whose best word score for automatic alignment (and using smaller databases but a larger test set) was just over 70%. Somewhat unexpectedly in view of preceding discussion, results for automatically-aligned lexical data (Table I(a)) are generally comparable to those for manually-aligned data (Table I(b)).

In all cases of error, the assembly of the incorrect pronunciation was traced during the operation of the program to ensure that the fault did not lie with our C code, but was a genuine failure of the analogy process. Dedina and Nusbaum's figure of 9% words incorrect amounts to just 6 errors, 3 of which are the same as ours. The commonest problem is vowel substitution (e.g. /ɛ/ for /i/ and /ɑ/ for /oʊ/, each on 4 occasions). In no case has confusion between /o/ and /oʊ/ transcriptions (see above) caused an error. It is possible to discount a very few of our errors as essentially trivial (i.e. those for *heen* and *feal*) reducing the error rate marginally to 20%. We conclude, therefore, that (see Damper and Eastmond, 1996) our re-implementation is incapable of reproducing Dedina and Nusbaum's quoted error rate of 9%. (Interestingly, Yvon (1996) has also failed to reproduce Dedina and Nusbaum's results, citing a word accuracy very close to ours of 72%.)

One deficiency observed with the simple shortest-path length heuristic is that the output can become unreasonably sensitive to rare or unique pronunciations. This is illustrated by the strange pronunciation of *mone* as /məni/, by analogy with *anemone*. A further example is the pseudoword *shead* which receives the bizarre, vowel-less pronunciation /ʃ_ _ _d/ (where '_' denotes the null phoneme) when using the D&N model, the TWB database, and manual alignment. As illustrated in Fig. 3, this turns out to be a result of matching the unique (but long) mapping *head* → /_ _ _d/ as in *forehead* → /fɔr_ _ _ _d/[6] (arc frequency 1) in conjunction with the very common mapping *sh* → /ʃ_/ as in *she* and *shed* (arc frequency 174) which swamps the overall score of 175. The same bizarre pronunciation does not occur with the PROD model. In this case, the path through the (/ɛ/, 3) node has a product score of $12 \times 30 = 360$ for the pronunciation /ʃɛd/ (taken by Sullivan, 1992, but not by Dedina and Nusbaum, to be a correct pronunciation) which considerably exceeds the score of 174 for /ʃd/. This particular problem does not occur with the Webster's database, which transcribes *forehead* as /fɔr_hə_d/.

**** FIGURE 3 ABOUT HERE ****

More generally, replacing the arc-sum heuristic of the D&N model by the arc-product as in the PROD model leads to a considerable increase in performance in all cases, e.g. from 77.1% words correct to 82.9% for the D&N 70 test set, Webster's database and manual alignment. The TP model performs better than PROD in all cases except for the D&N 70 test set, Webster's database, and automatic alignment, when it is marginally inferior.

Performance is generally poorer for all three models for the TWB database with the D&N 70 test set, perhaps because the TWB database has (as we suspect) been partly anglicised.

For the Sull 136 test set, our expectation of poorer performance (because of the slightly larger test set and/or inconsistency of dialect between the target pronunciations and the lexical database) is borne out for Webster's dictionary. For TWB, however, the performance difference between test sets is less consistent.

Overall, the two best results on pseudowords are 85.7% words correct for:

- the PROD model, with the D&N 70 test set, Webster's database and automatic alignment;
- the TP model, again with the D&N 70 test set and Webster's dictionary, but with manual alignment.

This rather poor performance offers no strong support for the position that PbA, at least as implemented here, is anything other than an inadequate model of the human process of reading pseudowords aloud. Perhaps a more explicit way of defining 'neighbourhood' than indirectly, by substring matching, would produce better results. For instance, Pollatsek and Rayner (1993, p. 415) conceive of a process whereby lexical entries in the neighbourhood of the input have their constituent phonemes activated, and the most "popular" phonemes are then somehow selected to assemble the output. Of course, an appropriate, computational definition of 'neighbourhood' remains problematic.

Lexical Words

Pronunciation by analogy has been mooted as a possible technique for use within a text-to-speech system. However, it is unlikely that the novel, 'out of vocabulary' input strings encountered in TTS applications will be much like the pseudowords used thus far in this area. Given this, we decided to test the ability of the various models to produce correct pronunciations for the entire set of lexical words. This is done by removing each word in turn from its relevant database, and obtaining a pronunciation by analogy with the remainder. In these tests, the transcription standard employed by the compilers of the dictionary becomes, as it were, its own reference. Hence, any problem of transcription inconsistencies between input strings and lexical entries is entirely avoided.

It could reasonably be argued that 'out of vocabulary' inputs are more likely to be typos, morphological variants of roots which are in the lexicon, proper names etc. (Indeed, one of the niches suggested for analogy methods is in deriving pronunciations for proper names, e.g. Coker, Church & Liberman, 1990; Golding, 1991; Golding and Rosenbloom, 1991.) Nonetheless, we feel that testing with a large set of 'real' lexical words is a step in the right direction.

*** TABLE II ABOUT HERE ***

Results for the testing of lexical words are shown in Table II. Again there is a consistent difference in the implementation models with the 'standard' D&N model performing worst and the TP model based on manual alignment doing best. In spite of uncertainty about its antecedents, all models perform better with the TWB database than with Webster's: this is probably simply a reflection of its slightly smaller size which means there are probably fewer exotic entries. Overall, the best result obtained with lexical words is 67.9% words correct (93.5% phonemes correct) with the TP model, TWB database and manual alignment.

For some lexical words, no pronunciation at all was produced because there was no complete path from *Start* to *End* in the lattice. This occurred for approximately 90 of the TWB words and 120 of the Webster's words irrespective of the specific model. We view this as a shortcoming: a TTS system should always produce a best-attempt pronunciation, even if it cannot produce the correct one. In some cases, this failure turns out to be a consequence of the specific form of the pronunciation lattice used by Dedina and Nusbaum, in which nodes are used to represent the 'end-points' of mappings. One of the inputs for which no pronunciation was found (for any model) is *anecdote*. The partial lattice for this word (manually-aligned database) is shown in Fig. 4. While not all arcs are shown, there is in fact no arc in the complete lattice between nodes (/k/, 4) and (/d/, 5) because there is no $cd \rightarrow$ /kd/ mapping anywhere in either dictionary. Nor indeed is there an *ecd* or *cdo* trigram – with or without the right end-point phonemes – which could possibly bridge the gap. This problem is entirely avoided with the Sullivan and Damper style of lattice, because the shortest-length arc corresponds to a single-symbol mapping rather than to a bigram (which may be unique). Thus, there will always be a 'default' single-symbol mapping corresponding to the commonest pronunciation of the letter. This is not to say that Sullivan and Damper's system will necessarily produce the correct output: it almost certainly will not because of the rarity of the $c \rightarrow$ /k/ mapping in the $\_d$ context.

**** FIGURE 4 ABOUT HERE ****

Another input which fails to produce a pronunciation is *aardvark*. The problem here is not that there is no *aa* bigram in the dictionary (which is found in words such as *bazaar*), but that it only appears towards the end of other words. Performing substring matching only over a restricted range (the number of matching comparisons is equal to the difference in length between the input string and lexical entry) is at the root of this problem. However, performing matching for all possible registrations of input string and lexical entry is likely to introduce problems of its own.

COMPARISON WITH RULE-BASED TRANSLATION

Is pronunciation by analogy a credible competitor to conventional letter-to-sound rules within a TTS system? This is a difficult question to answer directly for several reasons. Because, in principle, 'exceptions' to the rules can either be embedded in the rule set (i.e. treated as highly-specialised rules[7]) or segregated out and placed in a dictionary, there is a real problem in assessing the rules alone, in the absence of a dictionary, i.e. in deciding where the division between the two lies. Klatt (1987, p. 772) states: "A moderate-sized exceptions dictionary can hide the deficiencies of a weak set of letter-to-sound rules, but at a high cost in terms of storage requirements." According to Pols (1989, p. 60): "There is very little experience in evaluating the text-specific part (text preprocessing, grapheme-to-phoneme conversion) of a rule synthesizer." Further, Hunnicutt *et al* (1993) state: "direct comparison of performance of different systems is difficult due to the lack of standardized phone sets, data sets or scoring algorithms". Also, it is now standard procedure to test a data-driven technique with unseen , or held-out, data – not available to the system during training. It is not clear how this distinction applies in the case of a knowledge-based technique. That is, if a rule-writer devises a special rule for a problematic word (like *of* or *one*), should this be treated as 'seen' and excluded from the test?

In spite of these practical difficulties, system implementors still have to make an informed choice between the available approaches. Thus, we have felt it worthwhile to perform such a direct comparison.

The popular Elovitz *et al* (1976) rules were used for this purpose[8]. We are, of course, aware that this choice of rule-set might be criticised on the grounds that better ones exist. However, in line with the comment of Pols above, this is uncertain. Further, the Elovitz *et al* rules are at least publically available[9].

For the purposes of this paper, the comparison has been limited to testing on lexical words, using the TWB corpus. While the latter uses an inventory of 52 phoneme symbols, the rules produce output using an alphabet of 41 phonemes. The situation is slightly complicated by Elovitz *et al*'s use of both 'w' (as in *wear*) and 'WH' (as in *where*) in their phoneme inventory, presumably corresponding to /w/ and /ʍ/ respectively, as in some dialects phoneme of American English, whereas TWB uses only /w/. To cope with this, we have further reduced the inventory (to 40 phonemes) by collapsing these two. To some degree, the smaller inventory must favour the rules over PbA, since there is less opportunity for error. On the other hand, PbA assembles its pronunciations from the phoneme inventory supplied with the lexicon; this removes a potential source of error when scoring the output from a rule-based system (as one will not normally have perfect access to the rule-writers' inventory[10]).

The rules were used to transcribe the 16,280 TWB words, and the output scored against the TWB pronunciations, translated into the (smaller) phoneme inventory of Elovitz *et al.* The result was striking: 12,090 words were in error, corresponding to just 25.7% words correct, in comparison to figures of 64.5% and 67.9% for the best PbA implementations with automatic and manual alignment respectively[11]. Length errors (especially due to geminate consonants), /g/ ↔ /j/ confusions and vowel substitutions abound. Glushko's comment (1981, p. 81) that "… rules alone simply don't work" seems apposite!

Finally in this section, we note that (in spite of the shortcomings identified in out critical appraisal above), our most faithful implementation of PRONOUNCE still scores 63.0% words correct on the TWB corpus, vastly outperforming the rules.

CONCLUSIONS AND DISCUSSION

The ability of a PbA system, based on the PRONOUNCE system of Dedina and Nusbaum (1986; 1991), to derive correct prounciations for unknown words has been studied. The impact of detailed implementational choices on performance has been assessed, both for short pseudowords and for lexical words temporarily removed from the system's dictionary.

We find that, given the simplicity of the (monosyllabic) test sets, performance on pseudowords is poor at approximately 95% phonemes correct or 80% words correct for some of the better implementations. It appears that PbA – at least as implemented here and based on substring matching – is inadequate as a model of the human process of pronouncing pseudowords.

By contrast, PbA offers much promise for use within the pronunciation component of a text-to-speech synthesis system. In particular, it outperforms the popular set of manually-derived rewrite rules of Elovitz *et al* (1976) by such a margin that the latter can only be described as obsolete. Whether or not other, alternative rule sets can perform sufficiently well to be considered competitive with PbA remains an open question.

This work suggests several useful ways in which the performance of pronunciation by analogy might be enhanced. In general, results get better as the scoring method more closely approximates a probability estimate for the assembled pronunciation. Thus, it seems worth attempting to improve the approximation even further. It is also likely that the form of pronunciation lattice used by Sullivan and Damper, in which nodes correspond to the junctures between symbols rather than to the end-points of letter-phoneme mappings, could yield better results. On this point, Yvon (1996) recently re-implemented PRONOUNCE and obtained a word error rate of 56.6% on the Webster database (c.f. our value of 57.8%). Yvon then implemented an interesting improvement, which raised the accuracy to 64.0% (c.f. our best value of 60.7%), as follows. Dedina and Nusbaum's pronunciation lattice relies on there being an 'overlap' of exactly one character between matching substrings (i.e. the overlap corresponds to the nodes of the lattice). Yvon generalises this overlap to multiple characters, maintaining the requirement that the overlapping region for any two substrings must have the same corresponding phonemes before those substrings can be used to assemble a pronunciation. In principle, we prefer the direction of placing the nodes between letters, as this means the PbA system is a weighted finite-state transducer and makes the power of that formalism available to us.

We also intend in future to assess the impact of incorporating information about word frequency in the analogy process. Another rather obvious aspect of word structure ignored thus far is morphology. Hence, the current PbA implementations will match substrings across syllable and morpheme boundaries, whether this makes sense or not. Pollatsek and Rayner (1993, pp. 415–416) give the example of *velop*, saying that if this is to be pronounced by analogy to *develop* or *envelop*, "one needs guidelines to establish that the first two letters are a syllable, and the rest can be pronounced as if the syllable weren't there." Future work will concentrate on introducing knowledge of syllabic and morphemic structure into the analogy process, to determine if this can improve performance.

The impact of the alignment technique on performance was surprisingly small. Nonetheless, it would be worth undertaking a more thorough investigation of this aspect in the future.

It is worth asking if letter-to-sound rules, as traditionally employed in TTS systems, are really as bad as they appear here. That is, are the Elovitz *et al* rules at all typical? Pollatsek and Rayner (1993, p. 415) state:

> "... the pronunciation of English is too variable to sensibly be explained by rules, with estimates ranging from 25% to 75% of the pronunciation of words being accounted for by rules. (The percentage varies of course with the complexity and number of rules hypothesized.)"

so that our result is at the bottom end of this range. By contrast, Hunnicutt (1980, p. 74) indirectly estimates a performance of 71% words correct for her grapheme-to-phoneme and lexical stress rules, towards the top end of the range. Assuming (somewhat unreasonably) independent errors and (more reasonably) approximately 5 phonemes per word on average, this corresponds to a score of about 93% symbols correct – comparable to PbA. Hence, we feel justified in claiming that pronunciation by analogy works at least as well as the best letter-to-sound rules, with a tiny fraction of the development effort. This is consistent with the earlier finding of van den Bosch and Daelemans (1993) that "previous knowledge-based approaches were overkill" (at least for Dutch). It seems that rule-based systems only achieve acceptable performance because they are used in conjunction with a pronouncing dictionary (which does most of the work). Of course, PbA could be deployed together with dictionary matching in just the way that rules are, with the possibility of achieving better performance still. Indeed, in special circumstances (and assuming the rule-based approach retains specific advantages), PbA might even be deployed with rules (c.f. Golding, 1991; Golding and Rosenbloom, 1991): the two techniques are not necessarily exclusive.

Finally, a tabulation of human and machine pronunciations of the pseudowords can be found at URL:

`http://www.isis.ecs.soton.ac.uk/research/projects/anal.html`

including a sample of some of the erroneous pronunciations of the dictionary words, and other information on pronunciation by analogy.

Notes

[1]In the sense of being highly generalised, i.e. abstracted from the realisation of any particular word.

[2]Although Dedina and Nusbaum do not credit the source, PRONOUNCE bears similarities to the (apparently) earlier work of Byrd and Chodorow (1985).

[3]PRONOUNCE is designed to produce a single output, whereas Byrd and Chodorow's PRONUNC explicitly recognises that there is not always unanimity between subjects as to the pronunciation of a given pseudoword, and so produces multiple outputs.

[4]By *juncture*, we mean the abstract or virtual space between symbols. So, in the example word *h.e.l.l.o*, the junctures are marked '.'.

[5]It was obtained over the World-Wide Web from URL:

`ftp://ftp.cs.cmu.edu/afs/cs/project/connect/bench`.

[6]This is a highly unusual RP pronunciation, as is obvious from the correspondingly low arc frequency of 1. In fact, we do not agree with the dictionary transcription, which should in our view be /fɒd/. However, it would be highly dangerous in this work to treat the dictionary-transcribed pronunciations as anything other than sacrosanct.

[7]For instance, the Elovitz *et al* (1976) rules treat unique or difficult words like *of* and *one* using specialised rules of very limited applicability.

[8]They were obtained by anonymous `ftp` from directory `comp.speech/synthesis` at:

`svr-ftp.eng.cam.ac.uk`

[9]Rule-sets which have been developed as part of a commercial TTS system are not generally available in free-standing form, even to purchasers of the system.

[10]Elovitz *et al*, like others, give example words to illustrate the identity of their phoneme symbols. This does not, of course, define them unambiguously.

[11]This result is surprising in view of Elovitz *et al.*'s estimate of 66–69% words correct for their rules on a large corpus of general text, even allowing for the fact that this is an estimate. In view of the discrepancy, we are happy to make the UNIX script which we used for the evaluation available to other researchers who wish to verify our result.

REFERENCES

Ainsworth, W.A. (1973). A system for converting English text into speech. *IEEE Transactions on Audio and Electroacoustics*, **AU-21**, 288–290.

Andrews, S. (1989). Frequency and neighbourhood effects on lexical access: Activation or search? *Journal of Experimental Psychology: Learning, Memory and Cognition*, **15**, 802–814.

Baron, J. (1977). Mechanisms for pronouncing printed words: use and acquisition. In D. LaBerge and S. Samuels (Eds.), *Basic Processes in Reading: Perception and Comprehension* (pp. 175–216). Hillsdale, NJ: Lawrence Erlbaum.

Bosch, A. van den and Daelemans, W. (1993). Data-oriented methods for grapheme-to-phoneme conversion. *Proceedings of 6th Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, Netherlands, pp. 45–53.

Brooks, L. (1977). Non-analytic correspondences and pattern in word pronunciation. In J. Renquin (Ed.), *Attention and Performance VII* (pp. 163–177). Hillsdale, NJ: Lawrence Erlbaum.

Brown, P. and Besner, D. (1987). The assembly of phonology in oral reading: a new model. In M. Coltheart (Ed.), *Attention and Performance XII: the Psychology of Reading* (pp. 471–489). London: Lawrence Erlbaum.

Byrd, R.J. and Chodorow, M.S. (1985). Using an on-line dictionary to find rhyming words and pronunciations for unknown words. *Proceedings of the 23rd Meeting of Association for Computational Linguistics*, Chicago, IL, pp. 277–283.

Coker, C.H., Church, K.W. and Liberman, M.Y. (1990). Morphology and rhyming: two powerful alternatives to letter-to-sound rules for speech synthesis. *Proceedings of Eu-*

ropean Speech Association (ESCA) Workshop on Speech Synthesis, Autrans, France, pp. 83–86.

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

Coltheart, M. (1984). Writing systems and reading disorders. In L. Henderson (Ed.), *Orthographies and Reading: Perspectives from Cognitive Psychology, Neuropsychology and Linguistics* (pp. 67–79). London: Lawrence Erlbaum.

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

Damper, R.I. (1995). Self-learning and connectionist approaches to text-phoneme conversion. In J. Levy, D. Bairaktaris, J. Bullinaria and P. Cairns (Eds.), *Connectionist Models of Memory and Language* (pp. 117–144). London: UCL Press.

Damper, R.I. and Eastmond, J.F.G. (1996). Pronouncing text by analogy. *Proceedings of 16th International Conference on Computational Linguistics (COLING-96), Vol. 2*, Copenhagen, Denmark, pp. 268–273.

Dedina, M.J. and Nusbaum, H.C. (1986). PRONOUNCE: a program for pronunciation by analogy. *Speech Research Laboratory Progress Report No. 12*, Indiana University, Bloomington, IN.

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

Elovitz, H.S., Johnson, R., McHugh, A. and Shore, J.A. (1976). Letter-to-sound rules for automatic translation of English text to phonetics. *IEEE Transactions on Acoustics,* *Speech and Signal Processing*, **ASSP-24**, 446–459.

Federici, S., Pirrelli, V. and Yvon, F. (1995). Advances in analogy-based learning: false friends and exceptional items in pronunciation by paradigm-driven analogy. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI '95) Workshop on "New Approaches to Learning for Natural Language Processing"*, Montreal, Canada, pp. 158–163.

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

Glushko, R.J. (1981). Principles for pronouncing print: the psychology of phonography. In A.M. Lesgold and C.A. Perfetti (Eds.), *Interactive Processes in Reading* (pp. 61–84). Hillsdale, NJ: Lawrence Erlbaum.

Golding, A.R. (1991). *Pronouncing Names by a Combination of Case-Based and Rule-Based Reasoning*, PhD Thesis, Stanford University, CA.

Golding, A.R. and Rosenbloom, P.S. (1991). Improving rule-based systems through case-based reasoning. *Proceedings of American Association for Artificial Intelligence Conference, AAAI-91*, Anaheim, CA, pp. 21–27.

Humphreys, G.W. and Evett, L.J. (1985). Are there independent lexical and nonlexical routes in word processing? An evaluation of the dual-route theory of reading. *Behavioral and Brain Sciences*, **8**, 689–740.

Hunnicutt, S. (1976). Phonological rules for a text-to-speech system. *American Journal of Computational Linguistics*, **Microfiche 57**, 1–72.

Hunnicutt, S. (1980). Grapheme-to-phoneme rules: a review. *Speech Transmission Laboratory Quarterly Progress and Status Report, Royal Institute of Technology (KTH), Stockholm*, **STL-QPSR 2-3/1980**, 38–60.

Hunnicutt, S., Meng, H., Seneff, S. and Zue, V. (1993). Reversible letter-to-sound sound-to-letter generation based on parsing word morphology. *Proceedings of 3rd European Conference on Speech Communication and Technology (Eurospeech '93), Vol. 2*, Berlin, Germany, pp. 763–766.

Kay, J. and Marcel, A.J. (1981). One process, not two, in reading aloud: lexical analogies do the work of nonlexical rules. *Quarterly Journal of Experimental Psychology*, **33A**, 397–414.

Klatt, D.H. (1987). Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, **82**, 737–793.

Lawrence, S.G.C. and Kaye, G. (1986). Alignment of phonemes with their corresponding orthography. *Computer Speech and Language*, **1**, 153–165.

Lehnert, W.G. (1987) Case-based problem solving with a large knowledge base of learned cases. *Proceedings of 6th National Conference of American Association for Artificial Intelligence (AAAI-87)*, Seattle, WA, pp. 301–306.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory*, **10**, 707–710.

Lucassen, J.M. and Mercer, R.L. (1984). An information theoretic approach to the automatic determination of phonemic baseforms. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 84)*, San Diego, CA, pp. 42.5.1–42.5.4.

McCulloch, N., Bedworth, M. and Bridle, J.S. (1987). NETspeak – a re-implementation of NETtalk. *Computer Speech and Language*, **2**, 289–301.

Mohri, M. (forthcoming). Finite-state transducers in language and speech processing. To appear in *Computational Linguistics*.

Ogden, C.K. (1937). *Basic English*. London: Kegan Paul.

Patterson, K.E. and Morton, J. (1985). From orthography to phonology: a new attempt at an old interpretation. In K. Patterson, J. Marshall and M. Coltheart (Eds.), *Surface Dyslexia: Neuropsychological and Cognitive Studies of Phonological Reading* (pp. 355–359). London: Lawrence Erlbaum.

Pollatsek, A. and Rayner, K. (1993). Reading. In M.I. Posner (Ed.), *Foundations of Cognitive Science (2nd Edition)* (pp. 401–436). Cambridge, MA: Bradford/MIT.

Pols, L.C.W. (1989). Assessment of text-to-speech synthesis systems. In A.J. Fourcin, G. Harland, W. Barry and V. Hazan (Eds.), *Speech Input and Output Assessment: Multilingual Methods and Standards* (pp. 53–81). Chichester, UK: Ellis Horwood.

Seidenberg, M.S. and McClelland, J.L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, **97**, 447–452.

Sejnowski, T.J. and Rosenberg, C.R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, **1**, 145–152.

Stanfill, C.W. (1987). Memory-based reasoning applied to English pronunciation. *Proceedings of 6th National Conference of the American Association for Artificial Intelligence (AAAI 87)*, Seattle, WA, pp. 577–581.

Stanfill, C.W. (1988). Learning to read: a memory-based model. *Proceedings of Case-Based Reasoning Workshop*, Clearwater Beach, FL, pp. 406–413.

Sullivan, K.P.H. (1988). *The LaTeX IPA Fonts and their Usage*. Vision, Speech and Signal Processing Research Group Technical Note 88/TN1, Department of Electronics and Computer Science, University of Southampton, UK.

Sullivan, K.P.H. (1992). *Synthesis-by-Analogy: a Psychologically-Motivated Approach to Text-to-Speech Conversion*, PhD Thesis, Department of Electronics and Computer Science, University of Southampton, UK.

Sullivan, K.P.H. and Damper, R.I. (1989). *The Relevance of Psychological Models of Oral Reading to Computer Text-to-Speech Conversion*. Vision, Speech and Signal Processing Research Group Technical Report VSSP 89/TR1, Department of Electronics and Computer Science, University of Southampton, UK.

Sullivan, K.P.H. and Damper, R.I. (1990). A psychologically-governed approach to novel-word pronunciation within a text-to-speech system. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '90), Vol. 1*, Albuquerque, NM, pp. 341–344.

Sullivan, K.P.H. and Damper, R.I. (1991). Speech synthesis by analogy: recent advances and results. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '91), Vol. II*, Toronto, Canada, pp. 761–764.

Sullivan, K.P.H. and Damper, R.I. (1992*a*). Synthesis-by-analogy: a bi-lingual investigation using German and English. *Proceedings of International Conference on Spoken Language Processing (ICSLP '92), Vol. 1*, Banff, Canada, pp. 113–116.

Sullivan, K.P.H. and Damper, R.I. (1992*b*). Novel-word pronunciation within a text-to-speech system. In G. Bailly and C. Benoît (Eds.), *Talking Machines: Theories, Models and Applications* (pp. 183–195). Amsterdam: Elsevier (North-Holland).

Sullivan, K.P.H. and Damper, R.I. (1993). Novel-word pronunciation: a cross-language study. *Speech Communication*, **13**, 441–452.

Thorndike, E.L. and Lorge, I. (1944). *The Teachers' Word Book of 30,000 Words*. Columbia University, NY: Teachers' College.

Yvon, F. (1996). Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks. *Proceedings of Conference on "New Methods in Natural Language Processing" (NeMLaP '96)*, Ankara, Turkey, pp. 218–228.

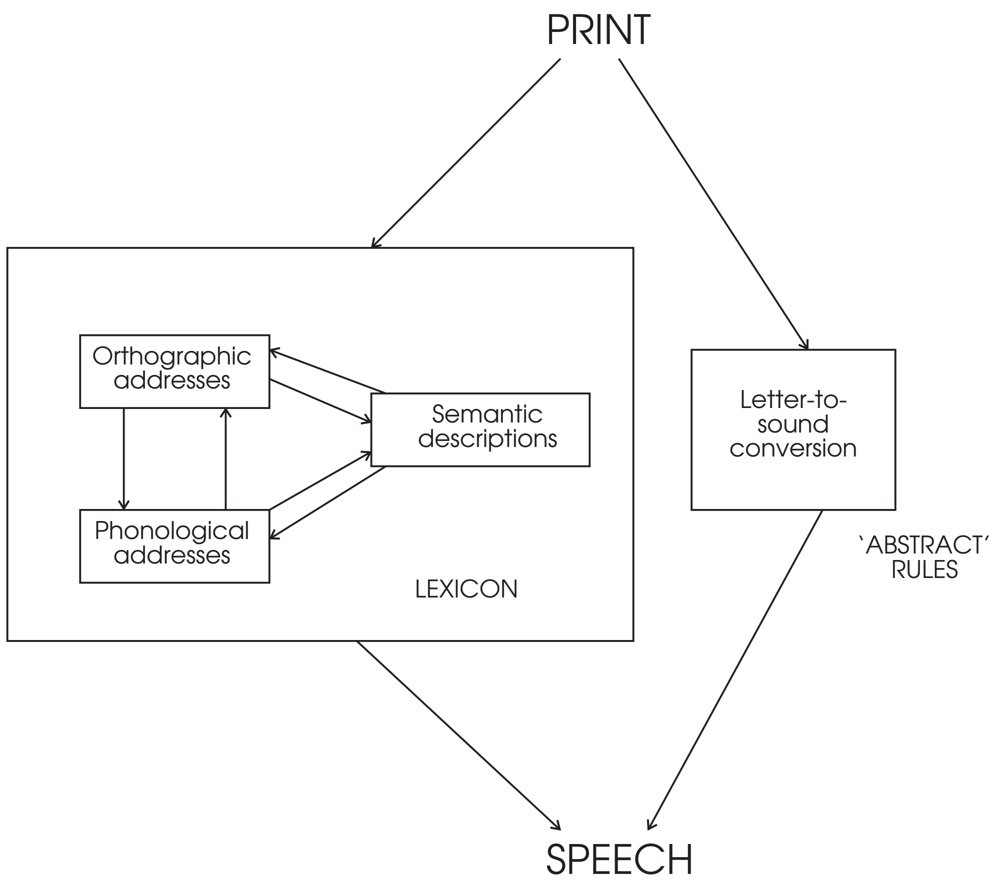

Figure 1: Schematic of the dual-route model for the production of pronunciations from print. The letter-to-sound rules are 'abstract' in the sense of being highly generalised, i.e. abstracted from the realisation of any particular word. After Humphreys and Evett (1985).

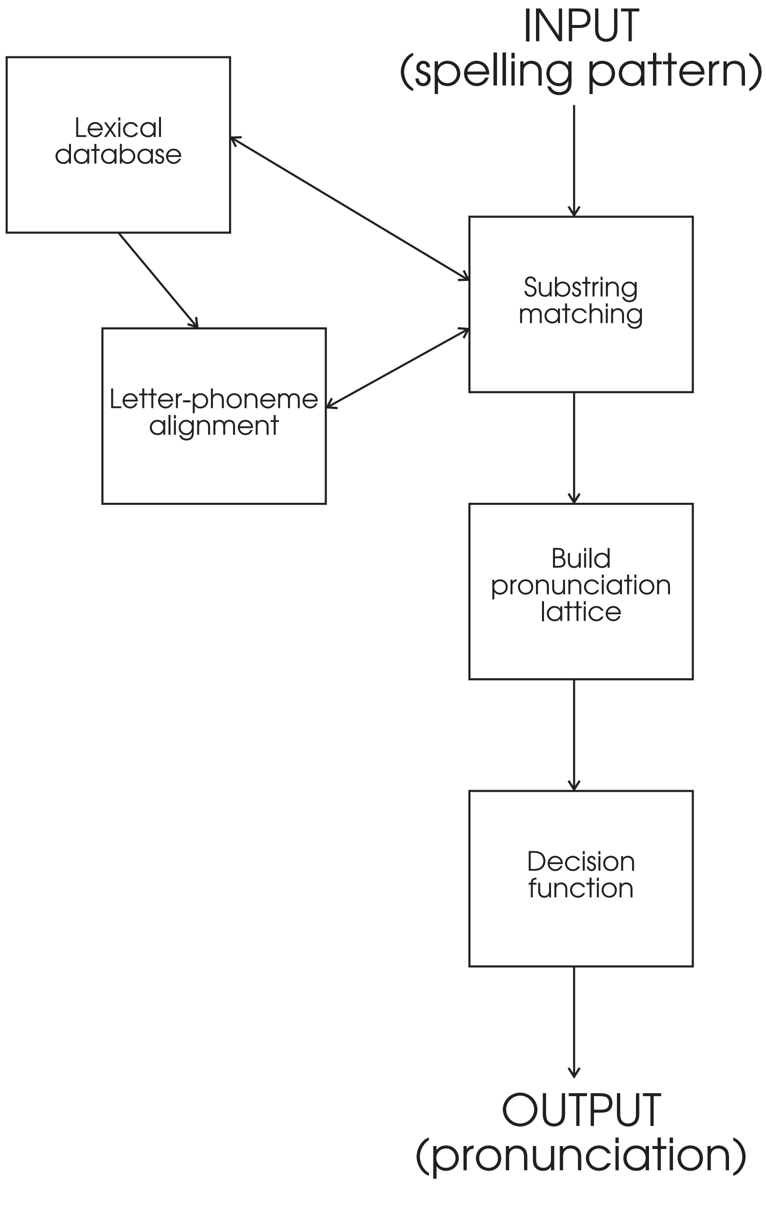

Figure 2: Schematic of Dedina and Nusbaum's PRONOUNCE.

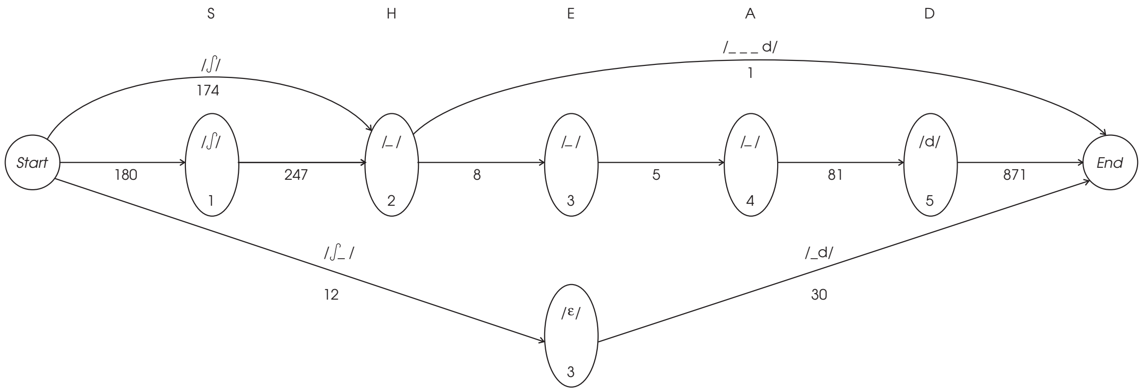

Figure 3: Simplified pronunciation lattice for the pseudoword *shead* with the *Teacher's Word Book* database (manually aligned). For clarity, only a subset of the arcs is shown and only for comparison of the input string with the lexical words *shed*, *forehead* and *head* (although the arc label values relate to the entire lexicon).

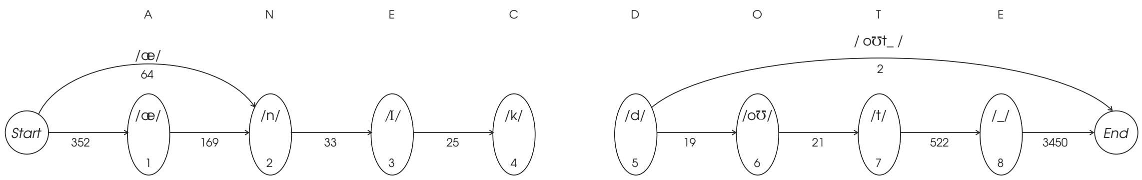

Figure 4: Simplified pronunciation lattice for the lexical word *anecdote* which fails to produce any pronunciation, since there is no *cd* → /kd/ mapping in the dictionary. While arcs are again omitted for clarity, there is in fact no arc between nodes (/k/, 4) and (/d/, 5).

Table I: Results for pronunciation by analogy of the 70 pseudowords used by Dedina and Nusbaum (1991) and the 136 word test set of Sullivan (1992). Lexical databases are *Webster's Pocket Dictionary* and *Teacher's Word Book*. In (a), the lexical data have been automatically aligned; in (b), they have been manually aligned. See text for specification of D&N, PROD and TP.

Test set	Implementation	Webster's (%)		TWB (%)	
		words	phonemes	words	phonemes
(a) automatic alignment					
D&N 70	D&N	75.7	93.9	68.6	91.6
	PROD	85.7	96.6	78.6	94.6
	TP	84.3	96.3	80.0	94.9
Sull 136	D&N	75.0	93.1	73.5	92.7
	PROD	82.4	95.2	77.9	94.1
	TP	85.3	95.9	79.4	94.3
(b) manual alignment					
D&N 70	D&N	77.1	94.3	70.0	92.6
	PROD	82.9	95.9	78.6	94.9
	TP	85.7	96.6	80.0	95.3
Sull 136	D&N	75.0	93.6	72.1	93.1
	PROD	80.1	95.0	76.5	94.5
	TP	83.8	95.9	81.6	95.7

Table II: Results for pronunciation by analogy of dictionary words for *Webster's Pocket Dictionary* and *Teacher's Word Book*. In (a), the lexical data have been automatically aligned; in (b), they have been manually aligned. See text for specification of D&N, PROD and TP.

Implementation	Webster's (%)		TWB (%)	
	words	phonemes	words	phonemes
(a) automatic alignment				
D&N	54.9	86.7	63.0	89.0
PROD	55.4	86.9	63.3	89.1
TP	57.0	87.1	64.5	89.3
(b) manual alignment				
D&N	57.8	90.4	65.6	93.1
PROD	58.5	90.7	66.1	93.2
TP	60.7	91.2	67.9	93.5