

**A Computer Tool for Use by Children with  
Learning Difficulties in Spelling**

**Helen Pain**

**Ph.D.**

**University of Edinburgh**

**1985**



I DECLARE THAT THIS THESIS HAS BEEN COMPOSED  
BY MYSELF AND THAT THE WORK DESCRIBED IN IT  
IS MY OWN:

## **Acknowledgements**

I would like to express my gratitude to the following people:

to my supervisors, Jim Howe and Ben du Boulay for comments, encouragement, and advice;

to Jane Hesketh and Brian Sharratt for typing, comments, proof-reading, and general support;

to Lincoln Wallen, Bernard Silver and Jeff Phillips for practical help with Scribe, references and printers;

to Alan Bundy for giving me time for writing, and to all the other people who gave me encouragement, advice, and comments, including Mike Sharples, Richard Young, Henry Thompson, Peter Ross and Graeme Ritchie;

to Roger Mitton for the corpus of spelling errors, to Arthur Biagi for judging them, and to Roland Backhouse for explaining his algorithm;

to Alan, Boog, Bob and Heather for putting up with me while I did it, and to Dave for the coffee;

and particularly to Mrs Wellwood, Mrs Raymond and the children of South Bridge Reading Unit, Edinburgh.

This work was supported by a Social Science Research Council studentship.

## **Published Paper**

A paper proposing the research described in this thesis was published in the proceedings of the W.C.C.E. (I.F.I.P) conference, (Lausanne, Switzerland) 1981: *Computers in Education*, R. Lewis and D. Tagg (editors), North-Holland. A copy of this paper is included, inside the back cover.

## Abstract

The development of a computer tool to be used by children with learning difficulties in spelling is described in this thesis.

Children with spelling disabilities were observed by the author, and their errors were recorded. Based on analysis of these errors, a scheme of error classification was devised. It was hypothesized that there were regularities in the errors; that the classification scheme describing these errors could provide adequate information to enable a computer program to 'debug' the children's errors and to reconstruct the intended words; and that the children would be able to recognize correct spellings even if they could not produce them.

Two computer programs, the EDITCOST and the PHONCODE programs, were developed. These incorporated information about the types of errors that were made by the children, described in terms of the classification scheme. They were used both to test the hypotheses and as potential components of a larger program to be used as a compensatory tool.

The main conclusions drawn from this research are:

The errors made by children with learning difficulties in spelling show regularities in both the phoneme-grapheme correspondences and at the level of the orthography.

The classification scheme developed, based on the children's errors, provides a description of these errors. It provides adequate information to enable a computer program to 'debug' the children's errors and to reconstruct the intended words.

Computer tools in the form of interactive spelling correctors are able to offer a correction for a substantial proportion of the child's errors, and could be extended to provide more information about the children's errors. They are also suitable for use with other groups of children.

## Table of Contents

1. Introduction	1
1.1. Children with learning difficulties in spelling: population and problem	1
1.2. Outline of the research	3
1.3. Questions addressed in this thesis	4
1.4. Plan of the thesis	5
2. Review of the Spelling Literature	8
2.1. Introduction	8
2.2. English Spelling	8
2.3. Cognitive theories of spelling and spelling disability	15
2.3.1. Theories of Spelling	15
2.3.2. Reading and Spelling Difficulties	20
2.4. Spelling errors	27
2.4.1. Debugging Errors	27
2.4.2. Spelling test and error classification schemes	29
2.4.3. Existing classification schemes	32
2.5. The Teaching of Spelling	38
2.5.1. Remedial Methods	40
2.5.2. Using Computers in Teaching Spelling	42
2.6. Computers and Writing	45
3. Review of the Computing Literature	47
3.1. Introduction	47
3.2. Detection v. Correction	47
3.2.1. Applications	47
3.2.2. Modes of Use – Interactive Programs	48
3.2.3. Use of context: syntax and semantics	49
3.3. Error Detection	52
3.3.1. Digram and trigram frequencies	52
3.3.2. Dictionary look-up	52
3.3.3. Dictionary size	53
3.3.4. Representation and search	54
3.3.5. What is a word?	55
3.3.6. Dealing with affixes	56
3.4. Error Correction	58
3.4.1. Previous errors and dictionaries of common misspellings	58
3.4.2. Matching substrings	59
3.4.3. Abbreviations	59
3.4.4. Measures of string similarity	60
3.4.5. Editing rules	62

3.4.6. Combined approaches	63
3.4.7. String to string repair	65
3.5. Summary	68
4. Educational and technical constraints of the correction programs	69
4.1. Introduction	69
4.2. Theory	69
4.3. Development of the error classification scheme and its relation to the correction programs	72
4.3.1. The children studied	72
4.3.2. The error classification scheme	73
4.3.3. Relating the spelling correction programs to the classification scheme	78
4.4. Program Design Constraints: Educational	80
4.5. Program design constraints: technical	82
4.6. A summary of the overall requirements	85
4.7. Failure of existing programs to satisfy the requirements	86
5. Design and use of a computer program to incorporate the editcost and phoncode correctors	92
5.1. Introduction	92
5.2. Example of a hypothetical session	93
5.3. Dictionaries	103
5.3.1. Size and content	103
5.3.2. Dictionary structure	104
5.4. Definition of 'a word' and affixes	104
5.5. Error Detection	106
5.6. Error Correction	106
5.6.1. Syntax	106
5.6.2. Semantics	108
5.7. The editcost program	108
5.8. The phoncode program	109
5.9. Other facilities	110
5.9.1. The lookup facility	110
5.9.2. The addword facility	111
5.10. Relating the design to the requirements	111
6. Testing assumptions about the use of the program	113
6.1. Introduction	113
6.2. Study 1	114
6.2.1. Introduction	114
6.2.2. Method	115
6.2.3. Results	117
6.2.4. Discussion	119
6.3. Study 2	121
6.3.1. Introduction	121
6.3.2. Method	122
6.3.3. Results	125
6.3.4. Discussion	130
6.4. Summary of results from the two studies	132

7. Detail of the editcost and phoncode programs	133
7.1. Introduction	133
7.2. Calculating the minimum cost repair: the editcost program	133
7.2.1. General overview	133
7.2.2. Shortlisting candidates from the dictionary: 'shortlist'	133
7.2.3. The editcost algorithm	136
7.2.4. Relation of the children's errors	141
7.2.5. Detail of the editcost program	145
7.3. Phonemic coding of words: the phoncode program	151
7.3.1. General overview	151
7.3.2. Related work	152
7.3.3. Design of the phoncode program	153
7.3.4. Defining the set of phonemes	154
7.3.5. Phoneme-grapheme correspondences	155
7.3.6. Phonemic coding of the dictionary	161
7.3.7. Detail of the phoncode program	163
7.3.8. Segmenting the word	163
7.3.9. Representing the dictionary	168
7.3.10. Matching the word and dictionary	171
8. Performance	173
8.1. Introduction	173
8.2. Performance of the editcost program	175
8.2.1. Testing editcost in use - Study 2	175
8.2.2. Testing on the corpus containing Study 1 and Study 2 errors	179
8.2.3. Errors which the editcost program failed to correct	180
8.3. Performance of the phoncode program	189
8.3.1. Testing on Study 1 and Study 2 errors	189
8.3.2. Errors which the phoncode program failed to correct	191
8.4. Results for combined programs	198
8.5. Results for individual children	201
8.6. Testing the programs on independent data	206
8.7. Summary	210
9. Conclusions	212
9.1. Introduction	212
9.2. Theoretical Questions	212
9.3. Practical Questions	213
9.4. Additional Questions	215
9.4.1. Information about the child	215
9.4.2. Providing information for the teacher	218
9.4.3. Use by other groups	219
9.5. Criticisms	219
9.6. Proposals for further research	222
9.7. Summary of conclusions	222
Appendix A. Editing commands, stories and notes on the children	224
Appendix B. Phoncode grammar and examples	246
Appendix C. Corpus of errors - both studies	248

## List of Figures

<b>Figure 1-1:</b>	Samples of children's writing	2
<b>Figure 1-2:</b>	Notation for vowel phonemes	6
<b>Figure 1-3:</b>	Notation for consonant phonemes	7
<b>Figure 4-1:</b>	Sample of population: chronological ages, spelling ages, and reading ages	74
<b>Figure 4-2:</b>	General classes of characters	76
<b>Figure 5-1:</b>	Example sections from the dictionary	105
<b>Figure 6-1:</b>	Words checked using the correction program	128
<b>Figure 6-2:</b>	Words selected when offered by the correction program	129
<b>Figure 6-3:</b>	Child realises the correct word is not presented	130
<b>Figure 7-1:</b>	First character alternatives for shortlisting	135
<b>Figure 7-2:</b>	Graph representing edit operation transformations	138
<b>Figure 7-3:</b>	Costgraph showing the node labelling	139
<b>Figure 7-4:</b>	Example of the spelling corrector in use	142
<b>Figure 7-5:</b>	Study 1: frequency of error types	144
<b>Figure 7-6:</b>	Study 1: frequency of types of spelling errors	145
<b>Figure 7-7:</b>	Example of candidates and minimum editcost	146
<b>Figure 7-8:</b>	Representation of 'special cases'	147
<b>Figure 7-9:</b>	Weightings assigned to edit functions	149
<b>Figure 7-10:</b>	Graph showing example weightings	150
<b>Figure 7-11:</b>	Examples of use of the phoncode program	153
<b>Figure 7-12:</b>	Representation of vowel phonemes	155
<b>Figure 7-13:</b>	Representation of consonant phonemes	156
<b>Figure 7-14:</b>	Finite State Grammar representation of 'caught'	164
<b>Figure 7-15:</b>	Alternative representation of 'caught'	165
<b>Figure 7-16:</b>	Finite state grammar representation of 'rime': v, c+e	167
<b>Figure 7-17:</b>	Finite state grammar representation of 'rime': v, c, e	167
<b>Figure 7-18:</b>	Representation of 'vc + e' graphemes	168
<b>Figure 7-19:</b>	Tree representation of part of the phoncode dictionary	169
<b>Figure 8-1:</b>	Editcost in use: outcomes of checking	176
<b>Figure 8-2:</b>	Editcost in use: individual results Correction offered	177
<b>Figure 8-3:</b>	Editcost in use: individual results Correction not offered	177
<b>Figure 8-4:</b>	Editcost tested on Study 1 errors	180
<b>Figure 8-5:</b>	Editcost tested on Study 2 errors	181
<b>Figure 8-6:</b>	Using editcost - Study 2 Errors for which correction not offered	183
<b>Figure 8-7:</b>	Testing editcost - Study 1 Errors for which correction not offered	183
<b>Figure 8-8:</b>	Testing editcost - Study 2, group 1 Errors for which correction not offered	184

<b>Figure 8-9:</b>	Testing editcost - Study 2, group 2 Errors for which correction not offered	185
<b>Figure 8-10:</b>	Comparison of judge's corrections with intended corrections - summary	186
<b>Figure 8-11:</b>	Comparison of judge's corrections with intended corrections - Study 1	187
<b>Figure 8-12:</b>	Comparison of judge's corrections with intended corrections - Study 2	187
<b>Figure 8-13:</b>	Phoncode tested on study 1 errors	189
<b>Figure 8-14:</b>	Phoncode tested on study 2 errors	190
<b>Figure 8-15:</b>	Comparison of errors corrected by the Phoncode program with those judged to be 'phonetic' - Study 1	193
<b>Figure 8-16:</b>	Comparison of errors corrected by the Phoncode program with those judged to be 'phonetic' - Study 2	194
<b>Figure 8-17:</b>	Errors judged to be phonetic, but not corrected: S1 and S2	195
<b>Figure 8-18:</b>	Comparison of errors corrected by Editcost and by Phoncode programs - Study 1	198
<b>Figure 8-19:</b>	Comparison of errors corrected by Editcost and by Phoncode programs - Study 2	199
<b>Figure 8-20:</b>	Testing of the editcost and phoncode programs on independent data - excluding repeats	207
<b>Figure 8-21:</b>	Testing of the editcost and phoncode programs on independent data - including repeats	208
<b>Figure A-1:</b>	Summary of Walter commands	225
<b>Figure B-1:</b>	Grapheme-Phoneme Correspondences: Grammar used in Phoncode Program	247
<b>Figure B-2:</b>	Examples of the phonetically coded dictionary	249
<b>Figure B-3:</b>	Examples parses of words according to phoncode grammar: alternative representation	250

## Chapter 1

### Introduction

#### **1.1. Children with learning difficulties in spelling: population and problem**

This thesis is concerned with children who, despite being intelligent, fail to learn how to spell.

Characteristically, they have spelling ages (as measured by standardized tests) which are two years or more behind their chronological ages. They are often not able to spell even common words correctly. In addition, they may have particular difficulty in specific areas: in perceiving patterns in words; in identifying rhymes; and in the sequencing of letters and numbers. The written work produced by the children is usually poorer than would be expected for their general level of ability. To demonstrate the extent of these difficulties, three examples of the writing produced by these children are given in figure 1-1. Commonly, these children have experienced earlier difficulties in reading, although some may show normal progress. Their reading difficulties are overcome more easily than their spelling difficulties.

In the literature they are referred to as children with learning difficulties in spelling, or with specific spelling disabilities, or more generally (particularly in cases where there is also a problem with reading) as 'dyslexic' (Ellis, 1981).

The major causes of the child's difficulty are often unspecified, or unknown. In some cases earlier language problems could be a contributory factor to learning difficulties, though causes such as deafness will have been ruled out. Emotional or personality problems have been experienced by some of the children: it is difficult to ascertain in these cases whether the emotional problems have caused the learning difficulty or vice-versa. Behaviour problems

to day i am waching the tahgfring i ask dad to waching the talhfi it is n on the talhfi at 7,15. am dad ask turno fr i askto am dad it is started. the hammer house horror the hammer house horror then i turno fr to waching the news started at 10 , oclock. (S.S. aged 10)

I live in a mansion it was bilt in 1717 many people were kild. I get nightmars this is what the nightmars is all about. It is winter and it it verey cold I alwas leave the window open. The butler opens the door and the man come in and gave the butler got some monye and shut the door and the man comeing up the stare she got to the top of the floor so I ran to my room. I was washing the mag from my room the man was the doctor I was supris a gasp come out of me. He pulld out a big sharp knif he went into my mum's room a squaek come from the floor bord's and then a skraem come from the room thiar was a big howl of wind I turned the crtane whigld and fell on the floor. I was going to scraem but i never. So i went to my mum's room i saw blode but not my mum. I went to my room and i got my case and packed and i ran away and i never cam back again (J.M. aged 11)

90 years ago in a lonly qorie in Midleton as work bigan some workers at the front talkin the then some one said I wonder how the old full got on with his gohst hunting they got to bottom look overthere thier was a body lay cut up one of us said hes warin a brass skurt hay hes a roman they did out what abuot my dad look heres a note join the legon at midnight then the body came together it said hulo its me dad remember to be here to night (L.B. aged 11)

**Figure 1-1:** Samples of children's writing

may also be evident. Part of the child's current lack of ability might be attributable to 'learnt failure': the child's teacher initially has a problem trying to convince him that he<sup>1</sup> may now succeed where he has always previously failed.

Traditional teaching has often failed to help these children. Even specialized remedial teaching has not always been successful. To help each child it is necessary to ascertain his specific areas of difficulty. Observation of the child at work, and collection and analysis of his errors, can provide the information needed to do this. However, acquiring this information is tedious and time-consuming for the teacher, given currently available schemes for classification of spelling errors. Additionally, this approach assumes regularity

---

<sup>1</sup>The children with learning difficulties in spelling are predominantly male: their remedial teachers are predominantly female. In general, the children will be referred to as 'he' and the teachers as 'she'.

in the child's errors. If these regularities can be shown to exist this information can be used to aid the child, both in the planning of remedial teaching and as a basis for supportive tools. These tools might also be used to motivate the child. They might take advantage of the child's abilities and compensate for his disabilities. There are few such tools designed to be used with this population: traditional tools such as dictionaries are of little use. Computer tools are advocated here as a means of enabling the children to communicate more effectively in writing. The development of a tool to be used by children with learning difficulties in spelling is described in this thesis.

## **1.2. Outline of the research**

Children with spelling disabilities were observed by the author, and their errors were recorded. Based on analysis of these errors, a scheme of error classification was devised. It was hypothesized that there were regularities in the errors. Two computer programs were developed, both to test this hypothesis and as potential components of a larger program to be used as a compensatory tool.

Initial testing and refinement of the programs were carried out. The two programs developed and implemented were both spelling correction programs. The first, the EDITCOST program, offered corrections for a misspelling by comparing it with potential corrections from a dictionary: those words that the misspelling could be altered to match most cheaply were offered. The cost of this matching was determined by the analysis of errors made by the children. Regularities noted in misspellings were used to reconstruct the intended word. The second correction program, the PHONCODE program, used information about the children's phonetic confusions to match the error to 'phonetically equivalent' corrections. Again, information about the regularities in the errors (in the phoneme-grapheme correspondences) was used to reconstruct the error.

Assumptions about the general methods of use of the programs were tested. The performance of the both correction programs was assessed: they were tested on data generated by two groups of children with spelling difficulties. Additionally they were tested on a corpus of errors generated by a sample of children of 'normal' spelling ability.

The two correction programs both used on-line dictionaries to provide the correction: this provided the means of retrieval of the word and compensated for the child's inability to do so. The child's problem became a recognition problem rather than a retrieval one. Other ways in which the program could be developed to help the child and his teacher were also considered.

### **1.3. Questions addressed in this thesis**

#### **Theoretical Questions:**

The following questions, relating to the children's difficulties with spelling, are addressed.

- \* Do the errors made by the children show regularity, or are they random?
- \* If a classification scheme is developed, based on the errors, can it provide adequate information to enable a computer program to 'debug' the children's errors and to reconstruct the intended words?
- \* Are the children able to recognize intended words (correctly spelt) even if they cannot produce them?

#### **Practical Questions:**

Questions relating to the practicality of a computer program as a tool are also considered.

- \* If a computer-based tool is provided to help the child:
  - is he able to use it?
  - is he willing to use it?
- \* How well does the tool succeed in correcting his errors?
- \* Do the answers to these two questions vary according to the individual using the tool?

#### **Additional Questions:**

Additional questions, not addressed directly, are also considered in this thesis:

- \* Could such a tool provide us with:

- information about the phoneme-grapheme correspondences used by individual children?
- other information that might be of use to the remedial teacher?

\* Could the tool be used by other groups of children?

## **1.4. Plan of the thesis**

Chapter 2: provides a review of the literature on spelling, including discussion of the theoretical psychological literature relevant to this thesis.

Chapter 3: provides a review of computer programs designed to detect and correct spelling errors.

Chapter 4: describes the classification scheme developed in this thesis; the way in which the program will be used in teaching; the requirements of the computer program that will incorporate the error classification and be used by the children.

Chapter 5: presents the overall design of a computer program, incorporating the spelling correction programs, and gives examples of the way in which it should be used.

Chapter 6: in this chapter two pilot studies are described in which various assumptions about the use of the program are tested. These studies are also used to collect data about errors.

Chapter 7: details of the EDITCOST and the PHONCODE programs are described in this chapter.

Chapter 8: the performances of the EDITCOST and PHONCODE programs are evaluated. The questions posed above are discussed in this chapter.

Chapter 9: conclusions drawn from the work in this thesis are summarized. Further work is proposed.

Conventions for representation of phonemes throughout the thesis text are given in figures 1-2 and 1-3.

International Phonetic Notation		
Examples	Alphabet(IPA)	used in text
late, day	eI	eI
air, care	ɛə	eE
bat, add	æ	ae
car, aunt	ɑ	a:
about, silent	ə	E
beat, keep	i	i
here, ear	Iə	IE
end, let	ɛ	e
maker, urn	ɑ	e:
ice, high	aI	al
ill, bit	I	I
boat, know	oʊ	EU
port, saw	o	o:
pot, soft	ɐ	O
food, rude	u	u
foot, book	ʊ	U
cube, unite	ju	ju
up, son	ʌ	^
oil, boy	I	ol
out, cow	aʊ	aU

**Figure 1-2:** Notation for vowel phonemes

International Phonetic		
Examples	Alphabet(IPA)	Notation used in text
bad, rub	b	<b>b</b>
bad, day	d	<b>d</b>
fat, rough	f	<b>f</b>
go, big	g	<b>g</b>
hit, behind	h	<b>h</b>
gin, joke	<del>dz</del>	<b>dz</b>
keep, cock	k	<b>k</b>
loud, kill	<del>l</del>	<b>l</b>
mad, jam	m	<b>m</b>
man, no	n	<b>n</b>
pit, top	p	<b>p</b>
run, bread	<del>r</del>	<b>r</b>
sit, loss	s	<b>s</b>
trap, step	t	<b>t</b>
very, love	v	<b>v</b>
wash, when	w	<b>w</b>
yellow, yet	j	<b>y</b>
zoo, beds	z	<b>z</b>
chair, lunch	tʃ	<b>ch</b>
ethics, accent	-	<b>ks</b>
quick, aqua	kw	<b>kw</b>
sing, along	ŋ	<b>ng</b>
sugar, bush	ʃ	<b>sh</b>
theatre, thank	θ	<b>th</b>
that, with	ð	<b>tv</b>
garage, pleasure	ʒ	<b>zh</b>

**Figure 1-3:** Notation for consonant phonemes

## Chapter 2

### Review of the Spelling Literature

#### 2.1. Introduction

In this chapter, the spelling literature relating to the work in this thesis is reviewed. The problems of English spelling, in particular the regularity of phoneme-grapheme correspondences, are discussed. Cognitive theories of spelling and spelling disability are presented and schemes of spelling error classification summarised. Methods of teaching spelling, and the more general use of computers in writing, are also considered. This chapter does not provide an exhaustive review of the literature, but is intended to give the reader an idea of some of the issues and problems.

#### 2.2. English Spelling

Writing systems in different languages represent speech at varying levels: lexical, morphological and phonological. Graphic representations may be used for words, syllables or phonemes. In Chinese, for example, a combined logographic and morphographic writing system is used: single graphic symbols may represent words ('logo') or morphemes (Stevenson, 1984). In Japanese single graphic symbols represent syllables, though some also represent morphemes. The majority of systems are alphabetic (or phonemic) (Hanna et al, 1966). Each graphic symbol in alphabetic systems represents one phoneme: the written system relates directly to the sound of the language. Ideally, in alphabetic languages, one grapheme would match to one phoneme and vice-versa. However, no systems exist that are quite so consistent: Finnish, Italian and Hawaiian come closest to true alphabetic languages. English is basically alphabetic, but the phonemes and graphemes do not translate one to one: it has only 26 single graphemes, whilst some 46 distinctive sounds are selected by many linguists as phonemes. So, combinations of graphemes are also used to represent phonemes.

The lack of consistent phoneme-grapheme correspondence in English has been blamed for difficulty in English spelling, and for reading difficulties. There is some evidence, however, that difficulties also occur in non-alphabetic languages such as Chinese and Japanese (Stevenson, 1984). There has been great controversy concerning the exact extent to which English can be considered an alphabetic language.

"Difficulties in spelling caused by inconsistencies in English orthography are primarily the result of an alphabet inadequate for the consistent spelling of necessary speech sounds and the historical impact of orthographies of other languages that have contributed to the English lexicon." p. 1282, (Horn, 1969).

The history of the English language gives us some idea of why the writing system lacks consistency. Many words in the language are of foreign origin: some of them have been 'anglicised' in speech without corresponding graphemic changes. When standardising spelling, early writers and printers disagreed on what those standards should be, and were inconsistent, (Horn, 1969). Additionally, spoken language continues to change: the written language must change also, or we must put up with the resulting inconsistencies. A third option is suggested by Skousen (Skousen, 1982): that the spoken language might change again in some way, to rematch the orthography. He suggests that there is a tendency to revive pronunciations which existed earlier in the language. In general, however, when changes occur in the spoken language, the orthography is left unchanged.

As well as varying over time, variations of the spoken language exist at any one time, in the form of dialects. Speech sounds are conditioned by the region in which one learns to speak, and also by socio-economic class (Groff, 1977). The orthography is not altered to match the dialect, except in direct phonemic transcription of speech. It could be argued, however, that whilst a particular phoneme pronunciation may be varied, the variation will be consistent and any phoneme-grapheme correspondences will be unaltered: a phoneme of one dialect will occur predictably in the same words in the same position as a different realisation of the same phoneme in another dialect (Cahen et al, 1971).

In considering correspondences between speech and orthography, the correspondences may go in two directions: from speech to orthography (spelling

or writing) and from orthography to speech (reading). Considering in particular phoneme-grapheme and grapheme-phoneme correspondences, the one cannot be considered to be the reverse of the other. Correct pronunciation of phonemes does not imply correct spelling of graphemes: Cronnell, cited in (Desberg, 1980) states that less than 50% of spelling-sound correspondence rules are reversible. Problems for spelling are not the same as problems for reading. For example, the schwa<sup>1</sup> presents little problem for reading if rules about reading unstressed vowels are known, though in spelling it is difficult to know which particular letter to use to represent it. 'Silent letters' also present a greater problem for spelling than for reading. A single phoneme may be represented by many different graphemes, and there are more different graphemes to represent each vowel phoneme than are used for each consonant phoneme (Groff, 1977). It is claimed that:

"Spelling-to-sound correspondences are with few exceptions uniquely defined whereas sound-to-spelling correspondences are not."  
(Baker, 1980)

Evidence to support this claim is provided by Venezky (Venezky, 1967), though he uses grapheme-to-morpheme and morpheme-to-spelling correspondences to relate spelling to sound. In reading the problem is of recognition: this may occur without phonetic coding. In spelling the problem is one of production: production is the more difficult process.

The question asked is "What evidence is there that there are regularities in spelling: in the orthography, or in phoneme-grapheme correspondences?"

Spelling reformists claim that there are too few correspondences, and too many irregularities; that generalisations about the relationship between spoken and written language are not useful. This is disputed.

Efforts have been made to reform English spelling by George Bernard Shaw, Isaac Pitman, Godfrey Dewey, and Theodore Roosevelt amongst others. (See (Venezky, 1980) for a short history of spelling reform in the USA). George Bernard Shaw is famous for his suggestion that 'if you took the 'gh' from

---

<sup>1</sup> the vowel sound occurring in unstressed syllables e.g. about, vowel, mother

'enough', the 'o' from 'women', and the 'ti' from 'nation' you could spell 'fish' as 'ghoti'<sup>2</sup>. Other examples of the lack of phoneme-to-grapheme correspondences are found in some of the different graphemic representations of the phonemes /el/ and /s/:

/eI/	they	hay	weigh	late	wait	fete
/s/	psychology	sitting	cite	confess		

The aim of spelling reform is to alter the orthography such that we have one-to-one phoneme-to-grapheme correspondence: an alphabetic writing system. Spelling would then cease to be a difficult and unpredictable task. Each sound would have a single graphemic representation. At first sight, it might seem a very reasonable idea to alter the orthography to directly represent the speech sounds. One might then ask, "Why has this not been done before?". Answers to this question, and reasons why spelling reform has not been successful, are given in the following paragraphs.

Taking the argument that history has caused spelling to be inconsistent, and that the orthography should be 'updated' to make it consistent: the spoken language does not cease changing, so we would be continually 'updating' the orthography to retain consistency. Current investment in printing alone makes any major reforms, even stable ones, impractical. A small number of minor reforms have been successful (for example U.S. spelling of 'behaviour', 'centre' as 'behavior', 'center') but these have had little effect overall.

Concerning 'dialect' arguments for spelling reform or alternative orthography, speakers of dialects such as 'Black English' (Desberg, 1980) have greater difficulty in spelling and reading because of the phonemic mismatch between dialect spoken and dialect represented in text. A solution might be to modify the orthography to teach 'Black English' speaking children initial reading and spelling, and later shift gradually to traditional orthography. An experiment was carried out in Britain in the 1960's that relates to this approach. Downing instigated the use of the Initial Teaching Alphabet (ita) in a number of British primary schools (Downing, 1965, see Groff, 1977). He devised the ita to provide

---

<sup>2</sup>This suggestion takes no account of the fact that 'gh' representing /f/ never occurs in the initial position in a word.

a regular grapheme-phoneme correspondence in words read. Children were taught initial reading skills, phonetically, using the ita, and later transferred to traditional orthography. There was a certain amount of controversy concerning the outcome of this experiment: significant differences were reported in favour of children using the ita (Downing, 1965) whereas others reported non-significant differences, and reservations concerning research design were raised (see (Groff, 1977) for more detail and discussion of this). It is not clear, therefore, that initial teaching 'in dialect' would be advantageous.

The reformist approach assumes little or no regularity in the orthography. This assumption has been questioned. In attempting to show the regularity of phoneme-grapheme correspondence in English (American English) Hanna and colleagues in Stanford analysed a corpus of 17,000 words, using a 52 phoneme classification (Hanna et al, 1966). For each syllable in every word they considered the corresponding phonemes; noting whether they were initial, medial or final; accented or unaccented. They found a correspondence between phonemes and graphemes, averaging more than 83% for the 30 consonant phonemes, and averaging 62% for the 22 vowel phonemes. The mean percentage correspondence for phonemes and graphemes overall was over 73%. They concluded that more than 49% of all words in the 17,000 word corpus could be spelled on a phonological basis alone, and that 80% of all phonemes could be represented by one grapheme. Various criticisms were made of this study: including, the pronunciation scheme used; the ad hoc designation of phonemes; arbitrary syllable divisions; use of inflected and non-derivational forms of words; and their failure to view English phonology as part of the total structure of English (Cahen et al, 1971). However, if morphemes had also been considered in this study, it is likely that less words would have been misspelt. Despite criticism of the Hanna study, Cahen et al (Cahen et al, 1971) suggest that there is evidence that phoneme-grapheme correspondences of some consistency do exist in English.

Clearer evidence for correspondence relating speech and orthography is shown in 'text-to-speech' studies (including (Venezky, 1966), mentioned above). Ellovitz, Johnson, McHugh and Shore used 329 letter-to-sound rules to convert English text to the International Phonetic Alphabet, with correct pronunciation for 90% of words and 97% of phonemes, in an average text sample (Ellovitz et

al, 1976). Ciarcia also used letter-to-phoneme rules to produce phoneme-based text, for a speech synthesizer (Ciarcia, 1982). Allen (Allen, 1981) initially analysed words into a set of morphs (= letter representations of morphemes), and letter-to-phoneme rules where the 'morph rules' fail, to produce speech from text.

Spelling reformists advocate revising the orthography to provide regular phoneme-grapheme correspondences: it is assumed that the object of the writing system is to reflect the sound of the language. Albrow (Albrow, 1974) refutes the necessity for involvement in pronunciation: he points out that there are a number of writing systems unrelated to sounds used in speech, for example, Chinese. He is not advocating 'learning every word by rote', however, but that correspondences relating to the deeper phonological system of the language should be looked for. Venezky (Venezky, 1966) also supports the belief that regularities in spelling do not always relate directly to the surface phonology, but may relate to an underlying form reflecting meaning. Baker (Baker, 1980) argues against spelling reform. He cites research by linguists showing that writing represents different levels of language, as speech does: that there may be regularities in the orthography in addition to phoneme-grapheme correspondences. For example, in the following three words, the final two letters are identical:

walked	/w/ /o:/ /k/ /t/
waited	/w/ /eI/ /t/ /I/ /d/
waved	/w/ /eI/ /v/ /d/

The 'ed' in each case indicates the past-tense morpheme. Phonemic transcription of each is given alongside. For each, the 'ed' is represented differently. If each were spelt phonemically (i.e. the orthography was reformed), then the morphemic information provided by 'ed' would be lost to the reader.

The orthography may also signal lexical derivatives of words. For example, in 'sign' the g is silent, apparently functionless: its presence, however, serves to indicate the relation between 'sign' and 'signal'. Additionally, homophones are often differentiated by different spellings: 'sign' and 'sine'; 'please' and 'pleas'; 'there' and 'their'. So, whilst phonemic spelling would remove some ambiguities, it would create others. Smith (Smith, P.T., 1980) also shows how the orthography indicates more than phonemic information, enabling decisions

about parts of speech, morphemic and syntactic structure. He focuses on the 'final silent e' to show how it can convey information from deep to surface levels.

Frith (Frith, 1980a) states that:

"Spelling in alphabetical scripts essentially means representing speech sounds. This spelling is visible phonology. However, spelling also reflects other levels of language. Plainly, correct spelling especially in English depends on other factors besides sound." p 2

Chomsky and Halle take an extreme view in claiming that English orthography is close to an optimal system for the English language (Chomsky and Halle, 1968). Whilst this extreme view is not held by the author, it is clear that the orthography should be considered in the ways in which it reflects underlying aspects of language: reflecting knowledge sometimes on the level of meaning or syntax, and sometimes on the level of sound.

## 2.3. Cognitive theories of spelling and spelling disability

A number of studies have focused on the orthography and difficulties of the English language, as a result of the influence of linguistics. Developments in Cognitive Psychology have caused a more recent shift towards consideration of the speller: his knowledge and processing capabilities. The main focus of this section will be cognitive theories of spelling, in particular theories of relevance to spelling disability. An excellent book, for further reading in this area, is 'Cognitive Processes in Spelling' by Uta Frith (Frith, 1980a).

### 2.3.1. Theories of Spelling

Early theories of spelling were associative theories. It was assumed that the correspondence between words, meaning and sound was learnt by rote, and memorised. Little use was made of generalisation (at the phoneme-grapheme, or at other levels) in spelling. Spelling was treated as a memory task. Failure in spelling was attributed to not having memorised a word correctly, or having forgotten it. It is difficult to see how the spelling of new unseen words or invented words would be explained by this approach. Otto, McMenemy and Smith (Otto et al, 1973) claim that attitudes and understanding are important, as well as memorising:

"Many students are faulty spellers because they have not learned to be careful spellers" p.256.

Whilst this view may seem a little harsh, motivation is obviously important. It should be noted that if a child has difficulties that are not helped he will lose motivation: he may become convinced that he is always going to fail, as he has done previously ('learnt failure'). However, whilst Otto et al suggest that 'he can't spell because he is uninterested', it may really be that 'he has become uninterested because he can't spell'.

Where spelling used to be considered a simple skill acquired by rote memorisation, it is now conceived to be a highly complex cross-sensory integration task (Hammill, 1975). Associative models are too simplistic: cognitive models have been developed more recently in attempts to explain spelling behaviour.

Valtin (Valtin, 1978) discusses different types of cognitive model of the reading process. This discussion is also of relevance to the spelling process. She distinguishes function models, process-orientated approaches, and task analysis approaches. Function models are based on the assumption that reading is the sum of a variety of cognitive functions e.g. visual and auditory discriminations, language skills, memory, and comprehension of symbols: if these functions are all sufficiently well developed in an individual, then the individual will be able to read.

"It is contended that undisturbed functioning of these complex operations guarantees reading achievement" p.214.

In particular, Valtin discusses these approaches in relation to their use in the study of dyslexic children. When using the function approach, groups of dyslexic children are compared with a control group on these functions. Low achievement on a particular function is taken to indicate a deficiency impeding progress. In remediation, deficient functions are trained by special programmes: reading should be improved by this training. Valtin criticises this functional approach on the grounds that there is no evidence for the transfer from the remedially trained function to reading, and that experiments with 'visual training programmes' have not been shown to lead to an increase in reading achievement. She also states that the approach is based on a fallacy:

"correlations are interpreted as causal factors, although the design of the studies does not permit this" p.214.

There is no evidence that the functions measured are relevant to the reading process (Valtin, 1978). These criticisms apply equally to the application of function models to spelling.

The second class of models distinguished by Valtin are subskill or task analysis models. The task analysis approach is concerned with the identification of the abilities and skills underlying successful spelling. Hammill and Noone (Hammill, 1975) describe a task analysis model of spelling put forward by Westerman; a hierarchy of abilities and skills that result in written spelling are described. These include basic language concepts, motor, visual, and auditory abilities, auditory-vocal-motor integration skills and visual-motor integration skills (see (Hammill, 1975) for more detail of this model). In relation to reading, Valtin cites work by Guthrie and Seifert (Guthrie and Siefert, 1977) which uses a

subskills approach, showing that the acquisition of letter-sound correspondence rules is similar for poor and good readers. They show that poor readers' subskills were comparable with those of normal (younger) readers of a similar reading level, but inferior to those of normal readers of the same age; also that specific tasks form a hierarchy of difficulty and could be used for diagnostic and remedial purposes (Valtin, 1978).

Continuing Valtin's discussion of methodology and models, the process-orientated approach tries to identify partial processes of reading in which dyslexic children are deficient. The concern in this approach is with the operations that take place in reading, e.g. feature analysis, segmentation, phonetic encoding and semantic decoding. Operations of this type are also applicable to the task of spelling. In process-orientated approaches, the development and speed of operations are studied, together with the degree to which they may become automatic.

Of these approaches the third, i.e. the process oriented approach, is of greatest interest in this thesis.

An example of a cognitive process theory is that developed by Simon and Simon (Simon and Simon, 1973). They state that the spelling of words can be produced in one of three ways depending upon the amount and kind of information about a particular word stored in memory. Highly over-learned and familiar words are learned via 'direct recall' in which the spelling of the word is retrieved directly, in response to the meaning of the word. Words for which only a partial image is found are spelled using a 'generate-and-test' process. Stored information about phoneme-to-grapheme correspondences is used to generate a plausible spelling for the word. This spelling is compared with stored visual recognition information, the partial image: if it matches, the 'test' succeeds. This information used in word recognition is accumulated in reading, and is used to aid spelling in testing the plausible spelling generated. For words that have not been seen before, and therefore have no stored information, a 'direct phonetic' spelling process is used: phoneme-grapheme correspondence information is used to generate a word, but there is no test process to verify it. Simon and Simon suggest that word-recognition and rule knowledge are being used to 'bootstrap' spelling competence. Whilst this is a

simplistic model, the 'generate-and-test' process is of particular interest in this thesis, and will be referred to again later.

Information used when spelling may relate to the sound, the written form, or the meaning of a word: that is, the information may be phonemic, visuo-orthographic, or semantic. An internal lexicon is conceptualised consisting of abstract word units having several identities, a phonological identity (information about acoustic, articulatory and phonemic properties of the word); an orthographic identity (information about the characters and their ordering in the word) and a semantic identity (knowledge about the meaning of the word) (Ehri, 1980).

Two strategies used to gain access to the information in the lexicon are:

1. the visuo-orthographic strategy - the access is direct and involves only the visuo-graphic code
2. the phonological strategy - the access is indirect, and the phonological code is generated by spelling-to-sound correspondences (in reading) or accessed by sound-to-spelling correspondences (in spelling)

Words with regular spelling can be read and spelled by either strategy. 'Irregular' words cannot be spelt using the phonological strategy alone, but necessitate the visuo-orthographic strategy, (Barron, 1980). Homophones also require access to the visual structure; non-words (i.e. invented spellings) can be spelt using phonological rules, but may also use visuo-orthographic information for spelling by analogy (Marsh et al., 1980).

Both phonological and visuo-orthographic routes can be used in spelling and reading, although those processes used in spelling are not simply the reverse of those used in reading (Nelson, 1980). In tasks where the direct visual route might seem to be faster and more efficient there is evidence that the indirect phonological route is also used. Baron (Baron, 1977) suggests that, even in cases where the visuo-orthographic path alone seems faster, both routes are used in parallel, facilitating each other. Cohen goes further to say that phonemic, visual and semantic analyses occur interactively. Whilst it may be assumed that years of reading practice promotes direct association between print and meaning, she suggests that phonological coding plays an important

part in reading for comprehension: that reading strategies are selected according to the demands of the task (Cohen, 1980). Baron (Baron, 1977) also provides evidence and argument supporting the use of the phonological path to meaning in reading tasks: that the phonological strategy is used as a device for remembering the wording of a sentence during comprehension.

If the lexicon may be accessed by either route in reading and spelling, why is spelling more difficult? It seems that, while reading allows direct access to visuo-orthographic information, this information is harder to access for spelling. Reading requires the recognition of words; spelling requires their production.

In order to find out more about word recognition, Henderson and Chard (Henderson and Chard, 1980) studied the 'word superiority effect': greater ease is shown in perceiving words than in perceiving non-words. Their conclusion is that recognition is mediated by an abstract graphemic representation, but that this graphemic code need not amount to sufficient specification of the spelling to aid production. So, whatever the reader's knowledge of the orthography, it may not help spelling: even if the knowledge were relevant, the speller may not be able to access it. Having information at a particular stage in processing does not mean it can be consciously retrieved (Henderson and Chard, 1980).

There is a commonly held belief that there is a generate-recognise loop available to supplement spelling recall. Relating to this, research has been carried out into imagery and spelling ability. There is some debate about whether imaging skills can be trained, and the usefulness of doing this. Sloboda claims that there is no relation between individual differences in imagery and spelling ability (see (Sloboda, 1980) for discussion). The conflict here is that 'imagery' implies a 'visible image of a word', but this does not necessarily mean a 'snapshot' representation: there is evidence that this 'visible image' does not exist (Henderson and Chard, 1980).

An alternative interpretation of the recognition-test loop view is that presented in the work of Simon and Simon (Simon and Simon, 1973) as mentioned above. This is criticised, however, for its strong emphasis on visual representation, and for their interpretation of the form of the visuo-orthographic information in the

lexicon (Henderson and Chard, 1980). The main issue debated here is, "is actual visual information used in recognition of words?" Henderson and Chard suggest that the information accessed in recognition is not purely visual, but that a robust phonological code also facilitates access to orthographic information. Tenney (Tenney, 1980) contends that spelling difficulties can be resolved by seeing which of two alternative spellings 'looks right'. She also claims that it helps more to actually see a potential spelling, rather than to imagine it: that there is the added advantage of the opportunity to make decisions based on phonological and linguistic judgements rather than on purely visual factors.

### 2.3.2. Reading and Spelling Difficulties

Early studies of learning and reading difficulties (generally referred to as 'dyslexia' by researchers) concluded that they were due to such causes as:

- \* sequencing disabilities, early language difficulties, indeterminate hand dominance, cross-laterality (Naidoo, 1972);
- \* eye movement control malfunction (Pavlidis, 1979);
- \* orientation difficulties, cerebral dominance (cited by (Miles and Wheeler, 1974) and by (Valtin, 1978)).

More recent studies have shown the results of these earlier studies to be both inconsistent and based on methodological weaknesses. Ellis and Miles (Ellis and Miles, 1978a) showed that the differences between performances of normal and dyslexic children were not due to faulty eye movements. Miles and and Wheeler (Miles and Wheeler, 1974) found no differences due to orientation difficulties. Valtin, disputing her own earlier findings, showed that difficulties were not attributable to visual perception, disturbed spatial frequencies, left-handedness, left-eyedness or mixed eye-hand dominance (Valtin, 1978). Dyslexic german students, learning english, were found to make no more errors than their peers, nor did they show faulty auditory perception (Jung, 1980). Nelson also found evidence that dyslexic children did not have particular sequencing problems, or letter orientation problems due to cerebral hemisphere domination; nor did they have visual perception problems or particular auditory perception problems as the root of their difficulties (Nelson, 1980).

The major criticisms of these earlier studies are of the methodology used. Valtin states a number of methodological weaknesses:

1. the diagnosis of dyslexia;
2. the criteria used to determine the extent of the population
3. selection of samples, especially with respect to age, sex, and degree of reading retardation;
4. biased selection of dyslexics (referred for secondary behaviour);
5. absence of appropriate controls;
6. lack of reliable and objective testing techniques

In identifying dyslexia, it is not easy to "clearly identify a group". Operational definitions have to be chosen, and criteria set for level of disability and level of IQ.

When comparing performance of dyslexics and non-dyslexics, a matched pairs design was used in earlier studies, matching for IQ. However, the IQ of good readers is, on average, higher than that of poor readers, according to IQ test result distribution. The matched samples are not therefore representative of their respective populations: they represent the top range of the reading disabled population, and the lower range of the good readers.

Additionally, intelligence quotient studies are given as single figures, though they include scores on various sub-tests. For example, the WISC full scale IQ includes a verbal IQ score and a performance IQ score. If full scale IQ's are matched, poor readers who have a weak verbal IQ will need to show relatively high performance IQ to match a balanced (i.e. comparable verbal and performance IQ subtest scores) 'good reader IQ'. It would be expected, therefore, that the poor readers would do better than the good readers on any test correlated with performance, and worse on verbal measures. These results cannot then be used to say that the population of poor readers are better at performance tests, and worse at verbal tests, than the population of good readers. Deficits shown by a group, therefore, can depend upon how pairs are matched. Deficits of reading disabled children, such as poor vocabulary, field dependence, and poor auditory discrimination, are said by Valtin (Valtin, 1978) to be "artefacts of the matched pairs design".

Later studies by comparison groups have tended to match on reading and/or spelling age, and chronological age, having two control groups: one control group is matched on chronological age, and the other on reading age, spelling age or both (Ellis and Miles, 1977; Nelson, 1980; Bryant and Bradley, 1980; Frith, 1980b). These studies have shown that there are similarities in the performance of children with spelling difficulties and spelling age controls, suggesting that there is some delay in development, rather than a fundamental disorder. Jung (Jung, 1980) showed that non-dyslexics have the same problems as children of a lower age, and that it takes the dyslexics longer to move from one stage to the next. Valtin (Valtin, 1978) cites a study by Guthrie (1973) who found that poor readers have the same subskills as younger children with the same reading age.

These more recent studies have suggested a number of causes of failure. Ellis and Miles believe that the dyslexic child typically suffers neither from visual nor from articulatory functional disorders, but from a lexical encoding deficiency. They have difficulty creating non-articulatory names or linguistic representations for symbolic language (Ellis, 1981). They say that in order to remedy this:

"What is in fact needed is the learning in a systematic way of grapheme-phoneme correspondences, with plenty of time being allowed for naming and for scrutiny of detail." p568 (Ellis and Miles, 1978b)

Bryant and Bradley showed that poor readers had difficulty in segmenting words. Valtin concluded, from the results of matched pairs studies, that dyslexia was one aspect of a broader learning disability, often coupled with auditory and speech deficiencies. However, she recommends further work using a case study approach, to attempt to isolate individual aspects of the reading and writing process, and to develop a theoretical model to include subskills and interactions between them (Valtin, 1978)

Three stages in the spelling process are hypothesized by Frith (Frith, 1980b):

- (i) correct analysis of speech sounds and derivation of appropriate phonemes;
- (ii) conversion of phonemes to graphemes by appropriate conversion rules, or by analogy;

- (iii) selection of conventionally correct graphemes from phonologically plausible ones.

Failure may occur at any stage in the process.

Studies by Bryant and Bradley, Nelson, R.Barron, and Frith (all described in (Frith, 1980a)) have focussed on poor readers and poor spellers, and their differing uses of visio-orthographic and phonological strategies. The findings of these studies will be discussed.

Bryant and Bradley (Bryant and Bradley, 1980) looked at the differences between poor and good readers, and younger and older readers. They showed that, initially, young and backward readers used a visual code for reading and a phonological code for spelling, and not the opposite strategies. Older readers and good readers, used phonological cues as well as visual cues in reading, and visual cues also in spelling. They concluded that, for the poor readers, it wasn't that they did not have these abilities, but that they did not know when to use them.

Nelson distinguishes two groups (previously bunched together under the heading 'dyslexic' or 'poor readers'):

1. children who are backward in reading and spelling - these will be referred to as PR/PS (poor reader/poor speller);
2. children who are backwards in spelling but show no reading deficit - these will be referred to as GR/PS (good reader/poor speller);

A third group would be normal readers and spellers - GR/GS (good reader/good speller). The difficulties of GR/PS cannot be ascribed to reading retardation. For the PR/PS group, Nelson says that the problem is one of acquisition of spelling knowledge by the semantic memory system: once acquired the information may be accessed by normal routes. The evidence shows that access by both routes is in keeping with spelling age level performance: neither route is impaired. For the GR/PS group, her findings show that they have phonemic analysis skills and knowledge of phoneme-grapheme equivalents, but that their ability to refer to specific word information and to decide on graphemic representation is impaired. The PR/PS group have a learning rather than a retrieval problem; the GR/PS have a retrieval rather than

a learning problem. She describes the problems of the first group PR/PS, as 'classic dyslexia' and of GR/PS as 'specific spelling retardation' (Nelson, 1980).

Barron (Barron, 1980) carried out experiments with good and poor readers (not distinguishing PR/PS and GR/PS). He suggests that children who differ in reading skills also differ in the predominance of visuo-orthographic and phonological strategies in reading and spelling. Poor readers rely on the visuo-orthographic strategy in reading and the phonological strategy in spelling, whereas good readers use both strategies. In trying to explain his evidence that showed good readers being more likely to use a visuo-orthographic strategy in spelling than poor readers, Barron hypothesises that there are inadequate visuo-orthographic entries in the lexicon. However, this would not explain why irregular words can be read, though not spelled: grapheme-to-phoneme rules are not being used for reading irregular words, therefore the visuo-orthographic information must be available. An alternative explanation he proposes is that visuo-orthographic entries do not have procedures for producing spellings, but only influence spelling directly in a checking process. This checking process might be one in which rule generated spellings are compared with the visuo-orthographic entries in the lexicon, and then correcting those spellings that fail the comparison test. His evidence suggests both explanations: he concludes that the poor and good readers differ in how they use the visuo-orthographic information in the lexicon during spelling rather than in the adequacy of the lexical information itself.

The last work to be summarised here is that of Frith, described in her book (Frith, 1980b). She describes work with three groups of children: GR/GS, PR/PS and GR/PS. The latter group she refers to as 'unexpectedly poor spellers'. Two questions asked were:

1. Do the groups GR/PS and PR/PS make different kinds of errors?
2. Do the GR/GS, GR/PS and PR/PS groups recognize words differently?

Both GR groups made more phonetic spelling errors than the PR/PS group, suggesting that GR/PS do use sound-to-letter rules. Both PS groups did equally well overall when tested on spelling single words. When spelling non-words, both GR groups used more phonetically acceptable spellings than

the PR group. However the GR/GS group used more 'favoured spellings' of the non-words (i.e. similar to Frith's own spellings), suggesting that the GR/GS group might also be using analogy in spelling.

With reference to the three stages in spelling proposed by Frith, she attributes the errors of the GR/GS group and most of those of the GR/PS group, (and some of the PR/PS errors), to the failure in the selection of plausible spellings stage. Most of the PR/PS failure is taken to be in the phoneme-grapheme conversion stage, relating to rule use : this would also be expected in the 'acquisition period of spelling' of younger children. Some of the failure of this PR/PS group may also be at the first stage: incorrect phoneme selection. This group have more phonological problems than the other PS group.

The two groups with normal reading ability, GR/GS and GR/PS, were also compared on reading strategies. It was found that the GR/PS group made more errors when reading aloud - 'by ear'; but no differences were shown when reading 'by eye', that is reading for meaning in text or from sight vocabulary. She suggests that the poorer spellers here are reading by partial cues and not attending fully to all the letters in the word. This would normally be an efficient reading strategy, but provides less opportunity for acquisition of underlying knowledge of spelling systems. Other tests - reading phonetically misspelt words, reading partially obliterated text, and selecting incorrect target words - supported Frith's partial and full cues interpretation.

In answer to the two questions posed by Frith, the conclusions drawn from this research were:

1. Good readers who are poor spellers make errors at the third stage of spelling, that is, they get plausible phonemic alternatives but can't decide which is correct. Frith suggests that they would benefit from learning 'letter-by-letter' structure of words and by careful reading based on full cues. Children who are poor readers and poor spellers make different errors, occurring at an earlier stage in the processing. They would benefit from learning phoneme-grapheme rules and phonemic analysis.
2. Good readers who are good spellers differ from those who are poor spellers in their reading strategy. The poor spellers do equally well at print-to-meaning, but worse at print-to-sound, or in reading nonsense words. Partial cues are efficient for reading, inefficient for

spelling: too much additional linguistic knowledge is lost. (Frith, 1980b)

General conclusions may be drawn from the research described here:

- (a) Good readers and spellers use both visuo-orthographic and phonological strategies in reading and spelling.
- (b) The visuo-orthographic and the phonological information in the lexicon are each accessed by separate routes, and in parallel.
- (c) Poor spellers and poor readers may favour a single strategy in reading or in spelling (or alternate strategies in each).
- (d) Visuo-orthographic information, which is accessible for reading by means of an abstract graphic code, is not always sufficiently well specified for spelling.
- (e) At times when this information is not retrieved for the production of spelling, it may still be used for the recognition of words, i.e. in checking spelling.
- (f) Failures in spelling may occur at several stages in the spelling process:
  - 1. in identifying phonemes;
  - 2. in conversion of phonemes to plausible graphemes;
  - 3. in the selection of correct graphemes.

## 2.4. Spelling errors

In cases where difficulties are being experienced with a skill, an obvious starting point is to look at the manifestations of these difficulties – at the errors or mistakes that are made in the performance of the skill. Data collected from the errors may indicate some pattern or rationale for them. This pattern, once identified, can aid theorising about the skill itself and the causes of difficulty. Once causes are identified remediation can be planned. The general approach of detect errors, classify them, infer causes, and remediate is well established.

When errors relating to the skill are fully specified, a variation on this approach may be used: the errors are reproduced from the description of the error pattern. If there are rules or regularities identifiable in the performance of the skill then the errors can be classified according to these. Errors may be attributed to an incorrect, incomplete or misapplied rule. If the error behaviour can be successfully reproduced from the classification of the errors, then to some extent the classification is validated. The approach of 'reconstructing the errors' has been taken in Artificial Intelligence research. In particular, it has been applied to the errors made in arithmetic, (Burton, 1982; Attisha and Yazdani, 1983).

### 2.4.1. Debugging Errors

Burton (1982) describes work by Burton and Brown in the domain of subtraction problems. The skill of subtraction is represented as a procedural network. Errors occur when correct subprocedures in the networks are called inappropriately or their results used incorrectly. These errors are referred to as 'bugs'. Student misconceptions are expressed as buggy procedures in the network. Student behaviour is modelled by a procedural network incorporating these buggy subprocedures (individually or in combination) which most consistently predict his errors on a set of problems. Closely related to this work is that of O'Shea and Young (O'Shea and Young, 1978) where bugs are accounted for in terms of incomplete production rules.

Brown and VanLehn extend the work described by Burton and Brown and propose a theory to explain these bugs (Brown and VanLehn, 1980). They do not attempt to explain (at a causal level) how and why the student initially

acquires these bugs, but propose a means by which buggy procedures can be generated. Repair theory suggests that procedures in the procedural network representing a skill are found to be inadequate, when used to solve certain subproblems. Consequently, a series of operators, representing some 'repair' to the procedure, are applied to the procedure to enable it to succeed. Bugs are thought to be caused by complex intentional actions, not by random unsystematic errors. For further detail, Brown and VanLehn's paper should be consulted.

The application of repair theory to other domains is also considered by Brown and VanLehn. They anticipate the applicability of the theory to other mathematical skills and, generally, to other procedural skills such as operating reactors, computer systems and traffic control. However, it is not clear that it would apply to spelling. Whilst procedures for arithmetic can be clearly specified, for spelling the rules and regularities are not so easily defined. There is some evidence that there are regularities in the spelling of normal spellers (see the discussion earlier in this chapter). Hanna et al. contend that there are regularities in spelling (Hanna et al, 1966). If regularities can be shown to exist in chronic misspellers, and these regularities identified, then they might be thought of as 'equivalent' to 'buggy subprocedures' in arithmetic. The rules and regularities used by the student to produce misspellings might also be used to correct these misspellings, to 'debug' the errors. These rules and regularities need to be identified in order to model the child's spelling skills. The focus here is on the former: on showing that there are regularities that can be used in debugging (correcting) a child's spelling errors.

There are other important differences in the tasks of debugging arithmetic and spelling errors. In the task of subtraction the (correct) expected solution is clear from the problem: the correction of an arithmetic problem can be calculated. In spelling it is more difficult to infer the intention of the speller - the 'problem' is not usually stated explicitly. The task of identifying and describing the error is more difficult. The initial concern of this thesis is to show that there are regularities that can be identified, rather than to model the child's knowledge of the spelling process. If sufficient regularities can be shown to exist, a further step would be to refine them into rules on which an individual model can be based.

In this thesis evidence will be presented supporting the argument that there are regularities in the spelling errors made by poor spellers. It will also be demonstrated that, for a large number of these errors, these regularities are sufficient to allow reconstruction of the correct word from the error: that spelling errors, like arithmetic errors, can be debugged.

#### 2.4.2. Spelling test and error classification schemes

In order to identify children with spelling disabilities, and to discover their particular difficulties, various tests and classification schemes have been devised. These will be described in general, and illustrated by particular examples, in this section.

Standardized spelling tests may be of two types. The first yields information about general spelling ability: children are given an age grade according to performance. These are generally used for comparative purposes and for identifying children experiencing difficulties. These are quantitative measures, giving spelling age, but no information about specific errors. Forms of these test vary: they can be tests of recall of spellings or tests of recognition. Brueckner and Bond (Brueckner and Bond, 1955) state that the former are better indications of spelling ability. In addition to standardized test, informal quantitative measures of spelling ability can be used for comparative purposes e.g. measures such as the number of misspellings per 1000 words that each child makes in general classwork.

The second type of standardized tests is designed to yield information about the specific skills and difficulties that the child has: it performs a qualitative analysis. Errors made are noted and classified according to type. The child's error tendencies, excessive and infrequent errors, are observed. Attempts are made to relate misspellings and phonic skills, mispronunciation and speech difficulties. These two types will be considered further: the former will be referred to as 'achievement tests' and the latter as 'diagnostic tests'.

The method of creating achievement tests is similar to that for creating other standardized tests. It is (roughly) as follows. A group of children of varying ages are given a set of words to spell. Words that are consistently spelt correctly by a set proportion (say, 80%) of children of the same age are noted:

The greatest age at which a word is spelt correctly by the majority is taken to be the 'age grade' of the word. Subtests of this set of words, each 'age graded', are then used to test other children: these children are each assigned the 'spelling age grade' of the words that they spell correctly.

Computers have also been used as an alternative means of test generation. Fremer and Anastasio (Fremer and Anastasio, 1969) developed a computer program to generate likely misspellings of words, given a set of replacement rules. Their aim was to produce spelling tests according to specifications about content and numbers of errors, i.e. to use the computer as an automatic achievement test item writer. They were interested in recognition type tests: students had to indicate which of a set of words and misspellings were incorrectly spelt. They developed a list of rules for misspelling (by looking at previously used tests) and generated sets of errors. A lot of plausible errors were created, but also many rules produced implausible ones. Recognition tests do not, however, require the use of the same cognitive abilities as the spelling recall tests. Gibson (Gibson, 1969), in a similar vein, used the computer to generate recognition type achievement tests. Using four edit operations, (deleting a letter, inserting a letter, changing a letter, or transposing two letters) she generated misspellings with 0 to 4 errors in them, and constructed sets of errors for recognition testing.

Some tests have been designed or altered to be administered by computer. One example is Hasselbring and Crossland's microcomputer version of the 'Test of Written Spelling' (an achievement test) (Hasselbring and Crossland, 1982). Their main objective was to determine whether examiner and scoring time could be reduced by using a microcomputer to present and mark the test. Students were presented with visual and auditory test instructions. Test words were presented (by audio cassette tape): the word was said in isolation; then said in a sentence; then said again in isolation. The student was then asked to spell the word, typing it into the microcomputer. The program waited until a response was made before presenting the next word. When the test had been completed the computer gave the tester a summary of the student's data, in terms of number of correct spellings and a list of errors made. They argue that examiner time is improved, and that there is less discrepancy in marking: human markers often disagreed in their interpretation of students' handwriting.

They claim that typed answers are more legible and that this is particularly important in the assessment of students who are disabled. The only information that their scheme gives, however, is whether each spelling is right or wrong.

The main problem with these graded achievement tests is that they yield a single score or spelling age grade. Peters (Peters, 1974) states that it is irrational for spelling tests to be of the same form as reading tests, especially normative reading tests.

"We are told e.g., that a child has a spelling age of 8.6 years, whatever this may be thought to reveal!" p. 56

Criterion reference tests are more useful, related to the skill of the individual child not to the performance of other children. Peters suggests that predictive questions should be asked, written work should be scrutinised and miscues noted so that spelling strategies can be identified.

Diagnostic tests provide more information about the child's difficulties. Errors are classed according to type, and information about the individual's abilities and disabilities is obtained. This information could be obtained informally. However, it would be an overwhelming task if carried out for a large group, although it can provide the teacher with insights that cannot easily be provided in any other way. Wedell, quoted in (Peters, 1975) defines diagnosis as:

"an attempt by systematic observation to arrive in a short period at information which could usually be obtained by informal observation over a longer period "

In diagnostic testing, errors are collected and analysed to determine if patterns exist in them. The errors may be taken from dictated material (text or word lists) or from uncorrected continuous prose produced by the child in the course of normal work.

In order to predict those types of words that would be difficult to spell, analysis has been made of misspellings of different types (for example the work of Mendenhall (1930) and of Gates (1937), cited in (Cahen et al, 1971)). These studies do not always indicate how frequently difficulties occur e.g. they indicate that words with diphthongs are harder to spell than those without – but not all words with diphthongs are hard to spell.

### 2.4.3. Existing classification schemes

A number of diagnostic test and classification schemes will be outlined briefly. The Gates-Russell Spelling Diagnostic Test 1940 is described by Hammill and Noone (Hammill, 1975) as having:

"more information about spelling than any other device reviewed."

It consists of a number of subtests designed to measure the ability to spell words orally, word pronunciation, ability to give letter-to-letter sounds, methods of word attack, auditory discrimination and other aspects. Each subtest is evaluated by grade equivalents and by recording brief descriptions of reactions and responses of the student, but no description of normative population is given (Otto et al, 1973).

Spache (Spache, 1940) suggests several common error types to classify errors made when spelling words from set 'lists of spelling difficulties'. The categories he suggests are:

- a. omission of
  - a silent letter
  - a sounded letter
  - a doubled letter
- b. addition
  - by doubling
  - of a single letter
- c. transpositions or reversals
- d. phonetic substitution
  - for a vowel
  - for a consonant
  - for a syllable
  - for a word
- e. non-phonetic substitution
  - for a vowel
  - for a consonant
- f. unrecognizable or incomplete.

Gentry (Gentry, 1978) provides a classification scheme designed to help teachers classify children's spelling strategies. Five strategies are given: for

each error attempts are made to classify it first in category (1). If that fails category (2) is attempted, then (3), (4), and finally (5).

1. correct strategy;
2. transitional strategy: the spelling is non-standard but looks like English; may be phonetically acceptable but incorrect, or may have letters right but in reverse order;
3. phonetic strategy: child uses letter names that best represent the sound elements of the word; may disregard legitimate letter sequences;
4. prephonetic strategy: essential sound features may not be represented, and letters omitted, but does contain rudiments of phonetic system;
5. deviant strategy: anything that does not fit above.

Simon (Simon, 1976) proposes the following classes of errors:

- \* errors of perception: including homophones, unknown words;
- \* errors of generation: phonetic misspellings, misapplication of spelling aids;
- \* errors of production: motor problem, handwriting;
- \* errors of checking.

Farnham-Diggory and Simon (Farnham-Diggory and Simon, 1975) propose four error categories:

- (i) sound frame errors - incorrect ordering of phonemes, missing syllables;
- (ii) phoneme errors - substitution of phonemes;
- (iii) grapheme errors - incorrect rendition of phonemes;
- (iv) visual frame errors - incorrect number of letters.

Nolen (Nolen, 1980) describes the errors in terms of rules by which child operates. Some examples of these rules are: a letter naming rule where 'e' represents /i/ or /e/; c and k alternate; rule for double letters depends on frequency and sound continuation of letters; syllables need a vowel (as in 'crumpul').

Peters (Peters, 1975) uses diagnostic dictations of about 100 words long, containing examples of deviant spellings. She states that it is unlikely that spelling is entirely random. She defines error classes and tries to assign to (a) first, then (b), and so on, (e) being 'unclassifiable'.

(a) substitution of letter strings:

- i. reasonable phonic alternatives
- ii. phonic alternatives not conforming to spelling precedent

(b) faulty auditory perception;

(c) perseveration: repetition of a letter string within the word;

(d) analysis of structure: omission, contraction, insertion, transposition or doubling of letter or letter strings;

(e) unclassifiable.

She defines three types of mis-cues:

(i) reasonable phonic alternatives;

(ii) phonic alternatives not conforming to spelling precedent;

(iii) random words.

Boder(1973) (cited by (Nolen, 1980)) defines three diagnostic categories:

1. dysphonetic: a deficit in grapheme-phoneme integration resulting in an inability to develop phonetic word analysis-synthesis skills; this includes extra letter errors and syllable omissions;
2. dyseidetic: a deficit in ability to perceive letters and whole words as configurations;
3. mixed dysphonetic-dyseidetic: letter sequences unrelated to target words, a virtual non-speller.

Boder developed a diagnostic screening test, consisting of ten lists of twenty, graded, phonetic and non-phonetic words, (Whiting and Jarrico, 1980). The child's sight vocabulary is established, and this is then used to test spelling. Time to spell each word is noted on the assumption that words spelled using phonological strategies take longer to spell. Different sub-groups are identified:

- (a) visual dyslexic: problems lie in confusion of the actual graphic symbols of the written structure of the alphabetic/phonetic script;
- (b) specific auditory problems: manifested by confusion and difficulties in the phonological route.

Similarly Moseley (Wade and Wedell, 1974) describes different types of spelling error. Some are purely visual - resulting from confusion between letters of similar shapes, from mistakes in left-to-right sequencing, or from a weakness in visual memory for letter-clusters and whole words. These cannot be explained in terms of poor auditory discrimination or indistinct speech. Visual confusion at level of letter-clusters frequently results from lack of attention to word structure. Also, he suggests, there are some problems of ability to discriminate sounds.

Schemes facilitating the reconstruction of the intended word from the misspelling are of special interest. Yannakoudakis and Fawthrop (Yannakoudakis and Fawthrop, 1983a) look for specific rules to help identify and correct spelling errors. They use the computer to compare correct and misspelt error forms and to test hypotheses about errors. They establish which are the common errors, or most likely ones, so that they can assign likelihood functions to possible errors that they later find, and then use this information in error correction. Three categories of error are defined: consonantal, vowel and sequential. Each splits into typical error forms, falling into one of three basic groups:

- (i) the grapheme is an obvious representation of the phoneme, referred to as an M element;
- (ii) the grapheme is a form which appears natural to the misspeller, referred to as an N form;
- (iii) all other forms are labelled L forms.

A number of consonantal, vowel and sequential errors are defined. They involve replacement of M and N elements by related M and N elements; doubling and singling of consonants; omission, addition, and replacement of similar silent/non-silent phoneme elements; transposition of adjacent and non-adjacent characters (see (Yannakoudakis and Fawthrop, 1983a) for more detail).

Generally, it appears that errors are classifiable according to whether they are:

phonetic or non-phonetic  
 transpositions or reversals  
 involve a letter (vowel or consonant)  
     a string or a word  
 whether letters are  
     omitted or added  
     singled or doubled  
     silent or sounded  
 unclassifiable

Classification in terms of omission, addition, or substitution alone mainly reflect only surface aspects of the language (Frith, 1978). Patterns of several letters or whole words need to be considered. The child's problem may be that he cannot segment the word into its component phonemes. In this case, classification in terms of omission/addition/substitution of letters alone is not helpful. The focus needs to be more on the individual child, rather than on the word alone. The actual phoneme-grapheme correspondences being used, and general rules relating to orthographic structure, need to be considered. However, at the other extreme, when errors are described in terms of high-level descriptions (e.g. Gentry's scheme, above) there may be insufficient information to enable classification of individual errors. Many schemes can be found to be tedious and time-consuming for a teacher to use if insufficient detail is provided.

In considering these schemes in relation to the theoretical issues discussed in the preceding section, none both:

- (a) relate directly to the three stage model described by Frith;
- (b) provide sufficient detailed description of the errors, in terms of visuo-orthographic regularities and phoneme-grapheme correspondences, that would permit the reconstruction of the correction from the error.

For many of the schemes their theoretical basis is not made explicit. In some cases function models (e.g. Moseley) or subskills models (e.g. Gates-Russell) are used, rather than process models. Some schemes, for example Spache's, give sufficient detail to enable reconstruction of the correction on a letter by letter basis, but provide no information to aid description of the errors in more general terms. Others, whilst relating to visuo-orthographic and phonological

strategies, give insufficient description of the error to enable classification of a single error into either class (e.g. Gentry, Farnham-Diggory). Decisions concerning the stage at which the child fails in the spelling process require more information than that provided in schemes such as those of Simon and Nolen. Boder and Peters's schemes rely on dictated words or text, and could not easily be used to classify errors made in free writing, and to debug a wide variety of errors.

Yannakoudakis and Fawthrop's scheme is the exception. However, the errors it is designed to deal with are those made by adults of normal spelling ability. Some critics have suggested that computers will never be able to "understand" the child's errors:

"Some children need a teacher to evaluate what they are doing when they misspell a word,...Computers are not yet wired for that kind of informed understanding." p539 (Nolen, 1980)

It is suggested in this thesis that computers **can** be used to provide and use information relating to a child's misspellings. Yannakoudakis and Fawthrop (Yannakoudakis and Fawthrop, 1983b) provide further evidence that a suitable classification scheme can be implemented in a computer program<sup>3</sup>.

---

<sup>3</sup>The description of their program was published after the author had developed the programs described in this thesis.

## 2.5. The Teaching of Spelling

In this section, issues relating to the teaching of spelling are raised. A number of schemes, including those involving computers to teach spelling and reading, are described briefly: remedial schemes are of particular interest.

In the teaching of spelling, arguments often revolve around the basic approach: whether it should be a 'whole-word' approach or a 'phonic' approach.

In phonics approaches, children are encouraged to look at the structure of words in terms of phonemes and corresponding graphemes. Phoneme-grapheme correspondence rules are taught and used for reading and spelling. In reading, sounds are associated with letters in the word, and these are 'put together' to build up to the whole sound of the word (from which the meaning is obtained). In spelling, the spoken word is broken down into its constituent phonemes, and these are each represented by appropriate graphemes. Explicit rules and other guides such as stressing, position in word, "what looks right", etc. are taught: these aid selection of the correct characters (graphemes) to represent each phoneme in any particular word. The 'Royal Road Readers' reading scheme is an example of the phonic word approach. Hughes (Hughes, 1973) provides a good general reference for, and introduction to, phonics teaching methods.

Phonics methods permit the student to tackle new words, but have the disadvantage of increasing the likelihood of spelling irregular words according to regular phoneme-grapheme correspondences. There is also some danger in phonics teaching approaches in concentrating on the sound and ignoring the meaning of a word, and in teaching differentiation and identification of letter shapes in isolation. Additionally, phonic methods have been criticised by those who believe that there is little regularity and consistency in the English language. This relates to the discussion of the extent and usefulness of phoneme-grapheme regularities in English earlier in this chapter.

In whole word methods (also called look-and-say or global methods) words are seen as whole patterns. The global graphic form is perceived and related directly to meaning in reading. In spelling, the complete letter pattern representing the word is generated from the meaning. In neither case is the

sound of individual elements used as an intermediary step. Perception of separate elements in the word is usually taught at a later stage. The initial aim is to build and extend the child's sight vocabulary<sup>4</sup>. An example of a whole-word scheme is the 'Janet and John' reading scheme. There is a danger here that, if individual letters are overlooked, spelling will be made more difficult and there will be more indiscriminate guessing. Additionally, if it is the sole method taught the child will not be able to deal with unfamiliar words.

Whilst there have been arguments over "which method is best" follow-up studies have generally shown little difference in the effectiveness of either. Further discussion can be found in Hanna (Hanna et al, 1966). It is more usual for both methods to be used in teaching, often starting with 'look-and-say', but introducing phonics methods early on (Hughes, 1973). The "Breakthrough to Literacy" scheme (MacKay, Thompson and Schaub, 1978) is a good example of an approach where the focus is both on the word as a whole (in composing sentences) and on its constituents (in building up and breaking down words).

The material to be used in spelling must be considered carefully. Several traditional spelling teaching methods have involved the rote learning of word lists. Hildreth (Hildreth, 1955) criticises this type of approach: the activity of spelling is not related to the rest of the curriculum; words are learnt out of context; words taught are not necessarily those that the child would wish to use; these words are often not related to each other.

If words are taught in isolation, the purpose of the exercise may be missed by the child. He may not relate the learning of lists to his writing in other subjects, or to communication in general. Even if spelling is memorized successfully when taught from a word list, if no context is provided (and no understanding of meaning of the word required) then there will be no association learnt between the spelling of a word and the object that the word represents. If spelling is taught in context it is easier for a child to learn association between the phonological, orthographic and semantic identities of the word. Whilst there are certain words that a child is expected to know, the 'frequency of use' counts used to select words may cause many words used

---

<sup>4</sup>Those words he can recognize immediately on seeing, without resorting to 'sounding out'.

frequently by individual children to be excluded from the lists (Bennett, 1967). Gathering material from a child's own needs and difficulties is more efficient than using published word lists (Harpin, 1976). Additionally, a child is more likely to be motivated to write, and to learn to spell correctly to do so, if the material is of interest and relevance to him (Clarke, 1982).

### 2.5.1. Remedial Methods

Ellis (Ellis, 1981) states that one feature of remedial schemes is that they should be centred on the individual. Weaknesses and disabilities of each child should be diagnosed. The teacher's main problem when planning teaching for individual children is diagnosing these particular difficulties (see the earlier discussion).

It might be that weaknesses are diagnosed in one modality: a child may have difficulties in auditory processing (perhaps in segmenting words) or in visual processing (for example, in sequencing letters in a word). In remedial teaching, the focus might be on the weaker modality (to strengthen it) or it could be on the stronger modality (to compensate) or on both. Hicks (Hicks, 1980), working with dyslexics with auditory processing difficulties, assessed whether it was better to teach to strengthen the preferred modality, or to improve the weaker modality. She tried three different training methods:

- training auditory skills (=weak skill)
- training visual skills (=strong skill)
- a combined approach

She found the best results were for the strong skill and combined skills approaches, thereby disputing the more usual assumption that the 'weak skill should be trained up'.

In relation to general methods, remedial approaches to spelling tend to be phonic (or synthetic) rather than whole-word. The initial focus is on the relationship between letters or syllables and sounds. The child is taught to be able to read and write regularly spelt words (Ellis, 1981). Three schemes will be briefly outlined.

A structured phonic approach is described by Wight-Boycott (Wight-Boycott, 1977). Words are organised into families based on similarities. The children

look at the construction of the words; they are read aloud; the child hears them and writes them down. The teacher helps the child to try and understand a few simple rules and observe similarities of sounds in words. The focus here is largely on individual words, and not on the presentation of the words in context.

"Alpha to Omega" is a useful handbook for teachers of dyslexics (Hornsby and Shear, 1978). It is a programme based on a phonetic, linguistic approach to teaching reading, spelling and writing. There is a strong emphasis on sounds and spelling patterns, and on phonic rules. Many examples of regular and irregular words, particular sounds, homophones, etc. are used. Some emphasis is placed on "seeing what looks correct" in spelling. Words are presented alone and in context, and there is concern with writing as well as reading and spelling.

The third example of remedial teaching is that taking place in a special Reading Unit, one of several in Edinburgh set up specially to help children with severe difficulties in reading and spelling. More detail about the structure and organization of the Unit, and of the children attending it, is given in chapters 4 and 6. A phonic approach is used in the Unit. Phoneme-grapheme correspondences are taught (or retaught) from scratch. The children are taught how to blend phonemes to build words. They are also taught to look for patterns in words, but warned to be careful of exceptions to rules. Emphasis is on the use of both visual and kinaesthetic modalities: they hear the word; write it; look at it; trace it; visualize it. A wide variety of reading schemes and teaching materials are used, including materials produced by the children themselves. The Unit is successful in increasing child's motivation, and successful in improving reading ability, but less successful in improving spelling ability.

It is not entirely clear why the Unit often fails to improve the children's spelling ability whilst succeeding in improving their reading. It is suggested here that the feedback that a child gets, relating to his errors, is an important factor. When a child makes a spelling error, and the teacher sees it, she provides feedback to him in information terms - he is told why he is wrong and how he might alter the error. He is not simply told "correct", or "not correct".

However, the teacher cannot watch every child all the time. Her criticism is positive but not immediate. If a child is unsure of the spelling of a word, (and is unable to use a dictionary to check it) unless he asks the teacher directly, he will have to guess. The child may repeat an error several times, without realizing it, before the teacher is free to check it. Unless teaching is on a one-to-one basis, it is not possible for the teacher to monitor and help each child continually. The tool described in this thesis is designed to help to solve this problem.

### 2.5.2. Using Computers in Teaching Spelling

In the last few years computers have become familiar objects in many schools. Software for teaching spelling and reading is now widely advertised<sup>5</sup>. However, evaluative studies of these programs in use are more rare. Few studies have examined the use of computers in spelling and in particular in remedial spelling. There has been some related research in the teaching of reading, using computers. Additionally, general arguments have been presented supporting the use of computers in Special Education (Howe, 1979). In this section a selection of computer programs for teaching spelling and/or reading are reviewed.

The programs that are probably the easiest to implement, and most prolific, are variations on drill-and-practice programs and hangman games. These are also probably the least sound, educationally. A number are listed in the software reviews of current magazines and journals e.g. Educational Computing, Journal of Learning Disabilities<sup>6</sup>. Special purpose hardware has also been designed for such programs, e.g. Texas Instruments' "Speak and Spell" and "Spelling B" (Zinn, 1978). Leveux (Leveux, 1977) advocates this method of teaching spelling though little evidence of success is presented.

Programs such as those of Grocke (Grocke, 1982), Candy (Candy, 1982) (and a number of those reviewed in Educational Computing magazine) involve spelling (typing) and reading single words. Grocke is concerned with the development

---

<sup>5</sup>It should be noted that this was not the case when the program of research, described in this thesis, commenced.

<sup>6</sup>Other more interesting programs are also listed in these reviews.

of reading skills in an educational clinic, focussing particularly on developing a sight vocabulary and on 'cloze-type'<sup>7</sup> writing exercises. Candy describes the use of two programs: the first for building a vocabulary 'pool' of the child's own words, and the second for the testing of the spelling of the words in the pool. In both studies an initial sight vocabulary is established.

Cumming, Terrell and Cassidy (Cumming et al, 1984; Terrell, 1984; Cassidy, 1984) each provide speech output with their programs. Cumming provides touch input to a speech system: the aim of the system is to diagnose children's weak and strong strategies. Terrell has several programs. His spelling program speaks a word for the user to type: if it is incorrect it does not appear on the screen - only correctly typed characters are echoed. He also provided programs for matching between a concept keyboard<sup>8</sup> and the spoken word, and for developing a sight reading vocabulary. Cassidy's program also uses speech to develop initial reading skills but takes a phonic approach. In these programs the children learn to associate the sound with the written word, in context.

Other programs involve typing words in response to pictures, or to generate pictures (Fiddy, 1982; Wilby, 1980; Prinz and Nelson, 1984). Fiddy (working with groups rather than individuals) teaches pre-reading skills, teaching letter-matching initially, then letter ordering and word input. Wilby's program teaches lists of words, using a phonic approach: if a word is typed correctly the appropriate picture appears; if it is incorrect, the correct word appears. The child's progress is monitored on-line. From the description of the project it appears that only nouns are taught. Association of a picture with a word does at least provide an object for the word to be related to, though there may sometimes difficulties when the same picture is interpreted as different objects (or given different names).

The final group of programs to be mentioned are those that focus more explicitly on phoneme-grapheme correspondences (Howe et al, 1978; Atkinson, 1974; Block, 1979) (though the first two are predominantly concerned with reading).

---

<sup>7</sup>Cloze tests, in reading, involve guessing single words omitted in sentences, using the context as a guide.

<sup>8</sup>Each key on a concept keyboard represents a word or concept, rather than a character.

Atkinson uses 'decoding' skills to teach initial reading. His program has different 'strands' each concentrating on a different aspect of reading. For example, strand 2 is a 'word strand' designed to improve the sight word vocabulary. The child is presented with a word in written and spoken form, and he is then asked to copy it. There is also a spelling strand in this program, which emphasizes regular phoneme-grapheme correspondences. Evaluation of the program's performance is described in the 1974 paper: children using it showed gains of 5.3 months in reading ability, compared with controls.

Howe (Howe et al, 1978) describes the teaching of phonic decoding skills to children with severe reading difficulties, working in a computer based environment. Results of the pre- and post-testing of the children suggest that handicapped children were able to learn decoding skills in this environment, and that there was transfer of learning to phonic skills used in the classroom.

Block's work (Block, 1979) is based on the generate-and-test process described in Simon and Simon, 1973. She evaluated the success of a teaching program based on this process. She used two main procedures in teaching. In the first, a tutorial program called SPLPAT (Spelling Patterns) was used to teach the children the various graphemes which may be associated with different phonemes, and that a given phoneme can have alternative graphemic representations. In the second, the children were encouraged to generate several plausible spellings for a word, and then to choose the correct spelling (testing each word to see if it was recognized). Studies of the children's performance on the programs revealed some support for the Simon theory. The children's performance on generating correct spellings improved, and more legitimate alternative spellings were used by the experimental group than by a control group. The children were limited in the words they used: it would be interesting to see if improvements generalised to writing outside of the experimental situation. The children in this study were able to generate a number of plausible spellings for a word, and to select the correct one more often than a control group. It is not clear how well this would work for children with spelling difficulties: it is likely that the correct word would be absent from those generated by the child. The general approach, however, is similar to that taken here.

## 2.6. Computers and Writing

Computer tools have been provided to aid writers of all abilities, styles and subject matter. A few of these tools have been provided with children as the target user. Claims have been made for the use of text editors with groups of children of average spelling ability (Clarke, 1982; Sharples, 1984):

- \* they allow easy manipulation of text;
- \* handwriting problems disappear;
- \* 'good copy' is easy to produce;
- \* drafts can be developed and altered without the need to show them in a "messy state";
- \* creative writing skills are facilitated.

Collins (Collins, 1982) provides an extensive review of research on teaching reading and writing with personal computers, including teaching beginning skills and reading comprehension; the use of writing aids and coaches for spelling and style; activity kits; text editors; automatic dictionaries and thesaurus; and general reading and writing environments such as publication, mail and information retrieval systems.

Quinsaas (Quinsaas, Levin and Gentner, 1980) briefly describes work on a writer's assistant. This would allow creation and modification of text, helping with low level skills such as spelling correction, and with higher level ones to aid and motivate writing.

The Writer's Workbench (Macdonald, Frase et al., 1982) also provides computer aids for text analysis, for use with adults. It includes the following facilities (though as it is still under development far more facilities may now be available):

### a. proofreading

- SPELLWWB - a spelling error detector
- PUNCT - for checking punctuation
- for checking consecutive 'same words'
- checks faulty phrasing
- looks for split infinitives

b. stylistic analysis

c. online reference and information

commonly confused words

SPELLTELL - a spelling corrector

on-line information about the program.

The SPELLWWB and SPELLTELL programs are described in chapter 3. These are designed mainly for use by adults with normal writing abilities.

The Quill project (Bruce and Rubin, 1984) is a set of micro-based writing activities for upper-elementary school children. It involves pre-writing, composing, revising and publishing aspects of the writing process and also a mailbag facility. It has been tested in school and showed significant improvement in the quality of the students' writing and in the writing processes and in their attitudes.

These tools are potentially very useful for children with learning difficulties in spelling. They have not yet been tested with this particular group, however, so no evidence is currently available to support the case for using them. They may also need to be modified in order to make them suitable. The writing skills of the children of interest here will generally be at a lower level than those of current users of these tools. The interface to the user may also need to be modified to accommodate children with spelling difficulties. How a tool might be developed, based on the facilities of the type provided by the tools described above, is described in chapters 4 and 5.

## Chapter 3

### Review of the Computing Literature

#### 3.1. Introduction

In this chapter the computing literature relating to the detection and correction of spelling errors is surveyed.

#### 3.2. Detection v. Correction

There is a distinction that needs to be made between programs that do error detection and those that do error correction. Error detection involves the isolation of misspelt words (the misspelling may even take the form of another word) or tokens in a text or program. Error correction involves taking a misspelling that has been detected and indicating a correction, or likely corrections.

Some computer programs are primarily concerned with detection of errors, and rely on the user to make the correction ('self-correction') for example: UNIX 'TYPO' (McMahon et al, 1978); UNIX/IBM 'SPELL' (Peterson, 1980a), and SPELLWWB (Macdonald, Frase et al., 1982). Conversely, a correction program can be used to correct an error once it is detected, or on request: e.g. SPELLTELL (Macdonald, Frase et al., 1982). Most spelling correction programs, however, carry out both error detection and correction.

##### 3.2.1. Applications

Spelling error correction programs have been used for a variety of applications:

- \* English prose e.g. 'writers' workbench' (Macdonald, Frase et al., 1982);
- \* children's interactions with the Plato educational system (Tenczar and Golden, 1962);

- \* text input by computer users e.g. DEC-10 SPELL (Peterson, 1980a);
- \* user interface applications, especially computer mail systems (Durham et al, 1983);
- \* large databases in Hebrew and English (Mor and Fraenkel, 1982);
- \* chemical abstracts (Pollock and Zamora, 1984);
- \* names of passengers in airline reservation systems (Davidson, 1962);
- \* retrieval of names from genealogical databases (Munnecke, 1980);
- \* spelling errors made in writing LISP programs, including misspelt keywords (Teitelman, 1978);
- \* errors made in writing SOLO programs (Lewis, 1980).

Of most relevance here are those concerned with the correction of spelling errors in English text, although other applications will also be considered where of interest.

### 3.2.2. Modes of Use - Interactive Programs

There are a number of different modes of use of programs. A complete file of text or code may be processed by the program, and then information about errors provided to the user: the UNIX SPELL program takes a file of text and outputs all words that it does not recognise; Pollock & Zamora (Pollock and Zamora, 1984) test a program to detect and correct errors in chemical abstracts; Atwell's correction program looks for errors in a large corpus of texts (Atwell, 1983). Alternatively, a program may be used interactively, permitting the user to check or correct a word at the time he writes it or as he proof-reads it: the Interlisp spelling corrector can intervene when the programmer makes an error, and can make a correction, (Teitelman, 1978); Durham, Lamb and Saxe are concerned with the correction of errors made as the user interacts with the computer; Peterson's spelling corrector (Peterson, 1980b) is designed to work interactively. The 'writers workbench' detector, SPELLWWB, takes a file of text and looks for errors, then permits the user to see each error in context and to use the corrector, SPELLTELL, to correct it interactively (Macdonald, Frase et al., 1982). The 'best way' of working - interactively or otherwise - will vary for different applications and for different users.

Different modes of operation are also offered. The Interlisp corrector operates in 2 modes: cautious and trusting. If it is able to correct a mistake, in trusting mode it will make the correction and continue as though no error had occurred. In cautious mode it will ask for approval before making the correction. Durham, Lamb & Saxe (Durham et al, 1983) discuss the difficulties of automatic correction of errors, particularly in the context of accidental acceptance of irreversible commands. Peterson (Peterson, 1980b) suggests asking the user whether he wishes to replace an error with the correction offered, or not, with the additional option of 'replace and remember' for future automatic replacement should the error recur. In the correction of commands in programming languages, automatic correction may be convenient for correcting mistypings, but dangerous when giving irreversible commands. In text, it may be desirable to correct all occurrences of a repeated misspelling, but less desirable when a correct word is categorised as an error, or when an error is miscorrected. When several different corrections are offered the user will need to be consulted.

### 3.2.3. Use of context: syntax and semantics

In considering computer methods of error detection and correction, an obvious starting point would be "How do we, as humans, detect and correct spelling errors?" If a word is misspelt, it may not 'look right' to us. We detect an error:

the boy closd the door

A word may have been misspelt in the form of another word and not 'make sense' to us:

the girl bounced the bull

Given errors in isolation, we can detect when a word is misspelt in the first way, but cannot detect when it is misspelt as another word. We need the context surrounding the word, at least, to recognise the latter sorts of errors though it may also be useful for corrections of the former kind. The difficulty of the task of proof-reading also demonstrates the strength of context: we often misread a word, in context, as we think it should be and may fail even to detect an error. Thus, in some cases, errors may be more easy to detect out of context.

One source of information which may help us to detect and correct errors in context is the grammatical structure of the sentence, or program code, in which an error is embedded: the syntax. Error detection in programming languages uses knowledge of syntax to detect and correct errors: the Interlisp correction program (Teitelman, 1978) attempts to correct mistakes by using the current context of the computation plus information about what the user had been doing. Programming commands are part of a well defined syntax and restricted vocabulary. Text, in 'natural language', is not so constrained. The study of syntax in natural language is a large area of research (Winograd, 1983). Research into 'ill-formed input' in natural language may give us some information about misspelt words in context (Fass and Wilks, 1983). It is difficult to discover what 'part of speech' a spelling might be labelled with, though some work has been done on this problem by Atwell (Atwell, 1983).

However, two problems, in particular, may arise:

1. What happens when the surrounding words, used to provide information about the misspelt word, are also misspelt:

I poot owt teh kat and karyd inn tha milke.

2. How do natural language programs cope with text that not only has misspelt words in it but that may also be 'ungrammatical' (or, at least, for which the grammar cannot be defined easily)? This can be the case with the children of interest here.

Additionally, in attempting to identify errors that are misspelt 'as other words' they may be of the same syntactic category as the intended word. Even if an error is detected, and we know that the correction should have a particular syntactic type e.g. noun, this still may not reduce greatly the number of possibilities for the correction.

When considering words in context, the meaning of the surrounding words and the general domain will also provide some useful information. Semantic analysis of the sentence or text may aid detection and correction of spelling errors. Fass (Fass, 1983) uses semantic information in spelling error correction. Again, as with syntactic information, it seems that the information provided here is not specific enough to aid correction. Research investigating the use of context to detect and correct spelling errors is not far enough advanced to allow formalisation of this process in a computer program, though current

researchers are starting to work on this problem (Mitton, 1984a). In the programs described in this thesis no context information is used to aid correction. The problem of detection will be further discussed in chapter 5.

One method of use of knowledge of the subject matter of the text would be the use of dictionaries specific to the topic area, to constrain the possibilities of options for correction. Pollock and Zamora use dictionaries specific to the type of documents they are concerned with. Peterson also recommends using dictionaries of 'document-specific' words. Use of dictionaries in error detection will be discussed below.



### 3.3. Error Detection

Different methods of spelling error detection, and issues specifically related to them, are discussed in this chapter.

#### 3.3.1. Digram and trigram frequencies

The UNIX TYPO program uses statistics of the frequencies of digrams and trigrams, in English, to detect probable errors. Typically, in a large sample of text, only 550 digrams (70% of possible digrams) and 5000 trigrams (25% of possible trigrams) actually occur, (Peterson, 1980a). The frequency of digrams and trigrams in the input text is computed. A list is made of distinct tokens or words in the text. For each token an 'index of peculiarity' is computed. This index is a statistical measure of the probability (for all trigrams in the token) that each trigram was produced by the same source as the rest of the text. TYPO outputs a list of tokens, sorted by index of peculiarity, highest index at the front of the list. Any token matching to a list of 2,500 common words is omitted and assumed correct. The user still has the task of deciding which of the words are really misspellings. This task is made more difficult by the lack of context.

#### 3.3.2. Dictionary look-up

The most common method of error detection is 'dictionary look-up'. Each input word or token is compared with a dictionary of legitimate words or a list of acceptable tokens. If a match is found then the word is considered to be correct. If no match is found the word is incorrect, or not in the dictionary. The UNIX SPELL program attempts to match each word in a text file with words stored in a large dictionary. The input words for which no matches are found are output as a list. Peterson's program also attempts to match input word and dictionary words, assuming an error if the match fails. Each word in a SOLO command (Lewis, 1980) is compared with a list of procedure names, and a list of node and relation names.

Authors of error detection programs, using 'dictionary looking-up', are concerned particularly with issues of size of dictionary, representation, and search strategies. (See (Peterson, 1980b) for a review).

### 3.3.3. Dictionary size

The larger the dictionary, the more likely it is that correctly spelt words will be matched, reducing the chance of correct words being rejected as errors. Pollock and Zamora use a 40,000 word dictionary to process some 25 million words (including repeats) (Pollock and Zamora, 1984). SPELLWWB (Macdonald, Frase et al., 1982) uses the UNIX 30,000 word dictionary. If the dictionary is too large, however, it may contain unusual and archaic words that could match to spelling errors, causing them to go undetected. Smaller dictionaries may constrain the vocabulary to include most words required by the user, but will increase the possibility of correct words not matching. The SOLO program checker (Lewis, 1980) uses a small table of stored words, but in this case it is desirable that the user is limited to a restricted vocabulary. Peterson (Peterson, 1980a), quoting statistics from the Brown corpus, states that over half of the tokens in normal English occur in a vocabulary of only 134 words. He also states that the total number of tokens in documents tends to be moderately small, 1,500 in a 10,000 word document, and that they often use words of particular interest to the subject area:

"for each specific document there exists a (small) table of words which occur frequently in that document (but not in normal English)".  
p. 681, (Peterson, 1980a)

He suggests a three-table structure of words to be searched:

- \* firstly, a small table of most common English words (100-200 words);
- \* secondly a table of (other) words already used in the document, constructed dynamically (1,000-2,000 words);
- \* finally, a large list of the remaining words in the main dictionary (10,000-100,000 words).

The approach of providing a dictionary containing vocabulary specific to the topic of interest, in addition to a small general dictionary, is taken in this study. It will be further discussed in chapter 5.

Efficiency of error detection depends upon the size of the dictionary, the representation, and the search strategy adopted. It also depends on whether the dictionary is stored in-core, or on a secondary disc or in a separate part of virtual memory requiring longer access and search times.

### 3.3.4. Representation and search

Input forms may be represented as strings of characters and matched to dictionary forms represented in the same way, (SPELLWWB (Macdonald, Frase et al., 1982), (Damerau, 1964)). Other representations may also be used. Davidson (Davidson, 1962) converts input forms to abbreviations to match to a list of names also stored as abbreviations. In order to speed up dictionary access, hash coding methods of representing words are used by Mor & Frankel (Mor and Fraenkel, 1982) and in the DEC-10 SPELL program (Peterson, 1980b). Mor and Frankel discuss the effectiveness of hash coding and make a case for its use. In the Dec-10 SPELL program words are initially coded by first 2 letters and word length, and each hash table entry is a pointer to a chain of words of the same initial 2 letters and word length.

Tenczar & Golden (Tenczar and Golden, 1962) use no 'string representation' of words. Instead, words are mapped onto a fixed number of bits, (the number of bits depending upon the word length of the computer, not the length of the word represented). Tenczar & Golden state that they use human criteria to set the bit representation of the word. Features of the word that are represented are length, first character, letter content, letter order (based on letter digraphs) and syllabic pronunciation. These features are ordered (subjectively) by importance. The coded input word is compared with a stored word and if no bits are in conflict the words are taken to be the same. Words that look alike are stored near each other in the dictionary. The dictionary is searched using 'binary chop search'. Details of the coding algorithm are not given, though the authors indicate that a scheme, using 41 bits to record information, has been shown to perform satisfactorily. Galli & Yamada (Galli and Yamada, 1968) verify the spelling of words by comparison with a 56,000 word dictionary, consisting of words, stems and affixes plus 2,000 common misspellings. The problems of affix analysis will be considered below.

Peterson (Peterson, 1980b) uses different representations for each of his 3 dictionaries. The dictionary of commonly used words is a modified 'trie'<sup>1</sup> structure, where the root is an array of all possible first letters. Branches go

---

<sup>1</sup>Peterson's terminology: not quite a tree.

for each of these first letter nodes to each 'next letter' node. The structure resembles a tree initially, with third level nodes holding third letters etc. However, all common endings are represented by the same nodes. So the tree becomes more like a graph. Information is stored in each letter node to indicate whether the letter is 'the end of a word' or not. Input words are matched, letter by letter, down the 'trie'. If no match is found, i.e. the end of the input word is not reached at the same time as a letter with 'end-of-word' marker, the document specific dictionary is checked. This dictionary is stored as a hash table, coded by first letter, last letter and word length. If the word is not found here, an attempt is made to match it in the very large disc-stored general dictionary. Here, words are represented as strings, but stored in blocks in direct access files. The words are sorted within the blocks, and blocks are indexed by the first and last word in each block. The correct block for an input word is found by binary search, and the word found by systematic search of the block itself. If the word is found, it is taken to be correct (and is temporarily moved to the document specific dictionary). If the word is not found it is taken to be a misspelling (Peterson, 1980b).

Two problems that have not yet been considered are:

1. What is a potential word?
2. How does a spelling detection and correction program cope with affixes?

### 3.3.5. What is a word?

In designing programs, decisions must be made about what are to be included as misspellings, and what not. How should the program deal with proper names, abbreviations and split or concatenated words? Should such characters as hyphens, apostrophes and digits be dealt with in the same way as other characters? Also, how should the problem of case be dealt with: should upper and lower case letters be considered as equivalent, or as distinct?

Peterson (Peterson, 1980a) addresses the problem of definition of potential words (tokens). He defines a word as:

"a sequence of letters, separated by delimiters"

He includes blanks and special characters, such as commas, full stops and colons, as delimiters, but points out the difficulty of interpreting numbers, hyphens and apostrophes. He suggests that the inclusion of numbers as letters should be left as an option for the user. He also recommends consideration of hyphens as delimiters, in normal use to create compound words, but not to permit 'end-of-line' hyphenation of words. Apostrophes are considered as letters when they are used as indications of omitted letters (I'd, won't) and as delimiters at the beginning or end of a token.

As Peterson states, the case problem is more complex. More thought is required when deciding how to deal with it when designing spelling correctors. Some proper names may be partly or entirely spelt in upper case, and might be considered misspelt if in lower case. When a word is capitalised at the start of a sentence or speech we will still want it to match with lower case versions elsewhere in the text: in this situation we may wish to ignore case altogether. However, we do not want to permit arbitrary (perhaps mistyped) capitalisation, occurring in the middle of words. In programming languages, the case of letters may carry more information: special decisions will have to be made to account for this.

Two words may be concatenated to make one, or a single word split into two. The detection algorithm may need to indicate this as an error. The SOLO program corrector (Lewis, 1980) looks for spaces missing in the input word: it matches the target command to the leftmost characters of the input word and then looks for a match for the rest of the character string. Pollock & Zamora's text correction program incorporates a special routine that checks for concatenation of 'function' words (i.e. non-context words in text, such as 'of', 'their', 'to'). Most programs do not specifically look for concatenated or split words.

### 3.3.6. Dealing with affixes

One method of reducing dictionary size is to remove all words with affixes (prefixes and suffixes) and store instead the root word and a list of affixes. Peterson (Peterson, 1980a) discusses affix removal. He suggests two approaches. The simplest involves examining the input token for affixes,

matched to a list of common affixes. These are then removed from the input taken and the stripped root matched with the dictionary. If a match is found the word is taken to be correct. Misspellings, where correctly spelt affixes are incorrectly attached to correctly spelt words e.g. carryed, babys, would not be detected. Peterson's solution is to flag each word in the dictionary with its legal affixes. Where affixes are stripped and a root word found, a flag will be checked to see whether the particular affix is legal for that root. The system of flags and interpretation, that Peterson describes (Peterson, 1980b), is that used in the Dec-10 SPELL program. Galli & Yamada (Galli and Yamada, 1968) claim that suffix analysis is not cost effective. Durham, Lamb & Saxe (Durham et al, 1983) suggest that affix analysis is not necessary in user-interface applications. Atwell, however, uses suffixes associated with particular sets of tags to provide information about text words not found in the dictionary (Atwell, 1983).

### 3.4. Error Correction

Having detected an error, by whatever means, the task of the correction program is to offer a correction, or possible corrections, for the error.

Methods of error correction are mostly based on some form of pattern matching of a word to a dictionary or list. Exceptions are those programs working at the sentence or phrase level, using syntax or semantics (Fass, 1983; Atwell, 1983), the difficulties of which have already been discussed.

Pattern matching in error correction is generally carried out in one of two ways. Either some representation of the misspelling is compared with a number of words in the dictionary, and the closest match (by some criteria) is taken to be the correction. Or, the misspelling is transformed in some way to produce a new word, and a check made to see if this new word is in the dictionary. Pollock refers to the former as a relative strategy and latter as an absolute strategy (Pollock, 1982).

#### 3.4.1. Previous errors and dictionaries of common misspellings

A simple method of error correction is to check whether the misspelling is in a list of previous spelling errors made. The Interlisp correction program (Teitelman, 1978) uses this as a 'first-pass' method: if it fails to produce a correction it then tries other methods. A similar method is that of comparing the misspelling with those in a dictionary of common misspellings. This may also be used as a 'first-pass' method (Pollock and Zamora, 1984) or in the case of the SPELLTELL correction program (Macdonald, Frase et al., 1982) the dictionary of 800 common misspellings is the main dictionary, against which 'relative' matches are made.

Techniques using relative strategies include:

- \* finding matching substrings in the misspelling and dictionary words;
- \* representing misspellings as abbreviations and matching to a dictionary of abbreviations;
- \* assigning scores to matching features of two strings to give a measure of closeness;

\* using a similarity key to represent features of an misspelling, as a code, and finding similarly coded dictionary words.

### 3.4.2. Matching substrings

To correct an error using the SPELLTELL program, (Macdonald, Frase et al., 1982), the user types in the part of the word that she knows to be correct. The program displays all the words in its dictionary in which the string occurs. The user then has to select the correct word from those matched strings offered: every string that matches is offered. For a user who is a reasonably good speller, and who is only unsure of the spelling of some small part of a word, this may be useful. However, should any part of this 'correct string' be incorrect the match will fail and the correction will not be found.

Alberga (Alberga, 1967) discusses an algorithm proposed by Baskin and Selfridge. The longest matching substrings of 2 strings are found and paired, then the longest matching substrings of the remaining elements are found and paired, and so on until no further matches can be found. The string with the greatest percentage matched to the misspelling would be the correction.

The SOLO spelling corrector (Lewis, 1980) also uses a 'matching substrings' algorithm. An initial check is made to see that there is not a difference of two letters or more in the lengths of the strings to be compared. The strings are then matched, letter by letter, left to right, then right to left. The number of matches is counted, and this number divided by the length of the larger string. If the resulting value is greater than 70% then a correction is assumed and reported to the user. If, however, more than one possible correction is found, no match is assumed.

### 3.4.3. Abbreviations

Davidson's program for finding names in a airline reservation system uses four letter abbreviations to represents surnames. Names are abbreviated by

1. deleting all vowels,
2. deleting all occurrences of H, W and Y (except where they are the initial letter),

3. deleting all but one of each set of contiguously replicated letters.

Blanks are used to fill the right end of the abbreviation if necessary. A comparison is made, with the list of abbreviated names, for an exact match. If no exact match is found scores are assigned to each pair of possible matches, where the score is one for each letter of the largest matching substring in the pair of abbreviations. If there are multiple matches the user can be asked to select manually.

Damerau (Damerau, 1964) describes a program by Blair that also systematically abbreviates dictionary items and words to be recognised. Taking the strings to be compared, they are each converted to a four letter abbreviation. A value is assigned to each letter in the string, proportional to the desirability of deleting that letter. The 'desirability' weighting is determined by the letter itself and its position in the word, and is an approximation of the probability of that letter in that position being an error. Those letters with highest 'probability of error' are systematically deleted from the string until four letters remain. The four letter abbreviations are then compared. If the abbreviation for an error matches to more than one four letter abbreviation from the dictionary then the process is repeated using longer abbreviations.

#### 3.4.4. Measures of string similarity

Correction algorithms proposed by Faulk (1967) & Morrison are described briefly in Alberga(1967). Faulk defines three measures of similarity between strings of arbitrary elements (sentences of words in this case). The measures are:

1. material similarity – the extent to which strings are composed of matched elements;
2. ordinal similarity – the extent to which the matched elements are arranged in the same order;
3. positional similarity – the extent to which the matched elements are located in corresponding positions.

Morrison, investigating the string matching problem in the context of C.A.I., proposed a number of simple approximations for matching strings:

- (a) the number of elements matched until the first mismatch;

- (b) the number of matched elements regardless of order;
- (c) the number of matches in increasing order;
- (d) the number of unmatched elements in each string.

A measure of closeness is used by Teitelman's Interlisp spelling corrector to find the closest match from a list to the error token. The 'closeness' is defined as being inversely proportional to the number of disagreements between two words and proportional to the length of the larger word. Characters in agreement are those, when the strings are scanned left to right, that are either the same or on the same teletype key, or lower case for upper case (or vice-versa). Transposed letters, in conjunction with no other letter disagreements, and doubled letters are not counted as disagreements. A criteria is set for closeness and if no word is sufficiently close an error is recorded, otherwise a correction is automatically made (or suggested to the user if approval is required). The corrector is restricted to Interlisp and uses lists for checking possible errors, the list used depending upon the type of the word being checked. Corrections of words selected as misspellings are moved to the front of the list so that, if an error is repeated, fewer comparisons will need to be made. Difficulty is encountered with setting a reasonable criteria of closeness for short words.

The Soundex system (Munnecke, 1980) uses a similarity key to represent names in a genealogical database. The representation of the names is based on pronunciation rather than spelling. Each name is given a four-character soundex code:

- \* the first letter of the name (the surname) is kept;
- \* all occurrences of vowels, and w,y,n, and g are deleted from the remainder of the name;
- \* the rest of the letters are assigned to numbered groups:
 

1 = b, f, p, v	2 = c, g, j, k, s, x, z	3 = d, t
4 = l	5 = m, n	6 = r
- \* if adjacent characters fall in the same numbered group, all but the first of that group are deleted;
- \* the name is then coded as the first letter followed by digits representing the remaining letters, in order;

- \* only the first 3 digits are kept, and if there are less than 3, zeros are added.

All matching names are found, from the coded list of names, and offered to the user.

In the Plato spelling correction program (Tenczar and Golden, 1962) features of words such as length, first character, and letter order are represented by a bit code. If the coded input word is compared with the dictionary, and no match is found, then the word is taken to be either a misspelling, or another word. The word with the least number of bits in conflict with the misspelling is found, and the number of bits in conflict is noted. If this number is less than some set value,  $k$ , then the word is assumed to be a misspelling. If, however, the misspelling differs, by more than  $k$  bits, from the closest dictionary word, it is assumed to be a different word i.e. it is not in the dictionary.

#### 3.4.5. Editing rules

Damerau (Damerau, 1964) indicates that 80% of all spelling errors are the result of:

1. transposition of 2 letters
2. one letter extra
3. one letter missing
4. one letter wrong

These rules form the basis of a number of spelling correction programs. The basic algorithm that they use is to construct a list of all the words (taken from the dictionary) which can produce the error by application of one of these rules. The list of candidate corrections is produced by multiple searches of the dictionary:

- \* testing for one extra letter in the misspelling involves deleting each letter from it, in order, and searching for the new word in the dictionary, a maximum of  $N$  operations for a word of length  $N$ .
- \* to test for transposition errors each pair of letters in the misspelling is swapped, and each new word produced is searched for: a maximum of  $N-1$  operations.

\* to test for one letter missing, or one letter wrong, involves a greater amount of searching:

- for a letter omitted a maximum of  $26(N+1)$  possible insertions<sup>2</sup>;
- for a letter wrong, a maximum of  $25N$  possible changes.

Any matches found are added to the list of candidates.

Peterson (Peterson, 1980a) suggests two strategies.

1. a 'match-any' character is inserted in all positions in the word (to test the one letter missing ) and substituted for all letters in turn (to test for a wrong letter), and each time a match is searched for;
2. each potential character is substituted (or inserted) in each character position, and search carried out normally.

The basic algorithm, using one of the two strategies suggested, has been used by the Dec-10 SPELL program (the first spelling corrector written as an applications program, 1971); by Durham, Lamb and Saxe (Durham et al, 1983); by Mor & Frankel, (Mor and Fraenkel, 1982) and by Peterson (Peterson, 1980b).

#### 3.4.6. Combined approaches

To further constrain search, digram and trigram frequencies can be used in conjunction with these edit rules. Omond, (Omond, 1977), uses digram frequencies to determine the order in which to apply the edit rules, and for which characters. Information about the frequency of occurrence of each letter-pair is used to select the letter most likely in error in the misspelling. The letter is replaced by the one most likely to occur in that position. The dictionary is searched for the resulting word. If a match is found it is accepted as the correction. Otherwise, the next most likely substitution is made. All substitutions are tried, then deletions and insertions, in order of letter likelihood using digram frequencies, until a successful match is made. However, the first successful match may not be the correct one.

---

<sup>2</sup>26 possible characters \* N+1 places it could be inserted

Zamora, Pollock & Zamora (Zamora, Pollock and Zamora, 1981) also use digram and trigram frequencies for spelling error detection, but conclude that they are insufficiently precise to be practical. Digrams are also applied by Cornew (Cornew, 1968) to single substitution errors.

Pollock & Zamora (Pollock and Zamora, 1984) also use the edit operations in their spelling corrector. A list of 'most similar words' is produced by comparison of the misspelling and the dictionary word, using two similarity keys. Pollock (Pollock, 1982) defines similarity keys as:

"techniques for generating compact representations of strings that preserve their fundamental properties but not minor details." p. 284

The selection of the actual correction is then determined by 'error reversal', a measure of 'distance' between two strings. The two keys that they use are firstly, the skeleton key, and secondly, the omission key. The skeleton key is coded by the first letter of the word followed by the remaining unique consonants in order, then the unique vowels in order. The omission key consists of all the unique consonants in the word ordered by those 'least likely to be omitted', with the least likely first, (see Pollock & Zamora, 1984, for frequency ordering); all unique vowels in the word are appended in order.

The list of 'similar' words found is taken and an attempt is made to transform the misspelling into one of these dictionary words by reversing one of the 4 basic edit operations or rules. If the attempt is successful the word is taken as a plausible correction. If there is more than one plausible correction the particular edit operation involved in each is considered. The focus is on 'the most likely reconstruction, given the error' rather than 'the most commonly applied edit operation'. Precedence is given to the operations in the following order: omission and transposition (equally); insertion; substitution. If more than one plausible correction still exist, then database frequencies of these corrections are considered and the most likely correction selected (Galli and Yamada, 1968; Morgan, 1970).

### 3.4.7. String to string repair

In the programs surveyed above, the four basic spelling error operations are only applied to misspellings containing a single error: transformations are created by a single application of one operation. This transformation does not, however, indicate a unique correction for each misspelling: a number of possible corrections can be produced from the same misspelling by a single transformation. e.g. the misspelling 'sall' can be transformed, by a single operation, into any of the following:

ball	call	fall	sail	sell	saul	sale
salt	tall	sally	stall	all	hall	wall

Referring to these edit operations, Pollock (Pollock and Zamora, 1984) states that:

"their usefulness lies in their correspondence to real world error-creating operations and their ability to interconvert any pair of strings." p. 361

By successful application of two or more error operations (or edit operations) any string can be transformed into any other string. For any pair of error and correction, there will be a variety of 'sets of operations' that can be used to correct one to the other. When testing to see which words can be interconverted in one operation, the maximum number of edits to be tried will vary, depending on the operation. To test for a transposition error, each pair of letters is swapped in turn, requiring  $N-1$  applications of the transposition operator (see previous section). The maximum number of operations required to test for transposition or insertion errors (i.e. by deleting each letter of the word, in turn, and checking for a match) is small and dependent upon word length. For substitution and omission errors, a far greater maximum number of operations is required to test for possible corrections. Greater time is taken by string-distance measurement techniques involving substitution and omission errors, especially where there are a large number of alternatives to be examined.

When more than one error is involved the maximum number of operations necessary, to test for all possible corrections, increases sharply. The number of plausible corrections found also increases dramatically, and the likelihood of the 'correct' one being found decreases. Pollock & Zamora conclude the following from this:

"Correcting multiple errors through these error reversal techniques produces a plethora of possible corrections with low plausibilities - a poor prognosis unless the parent vocabulary is extremely small. This suggests that the error correction technique should not be too powerful or it will generate too many false corrections". (Pollock and Zamora, 1984), p.361

However, what Pollock & Zamora do not consider is the possibility of controlling the application of the edit operations in such a way that 'more likely error' edits are considered first. This is done using digram frequency information, for single errors only by Omond (Omond, 1977). The possibility of weighting the operators when applying them to particular letters has also been considered (Morgan, 1970).

There exists a substantial literature on 'string-to-string' repair theory, in which repair between strings (i.e. transformations involving edit operations) involving multiple operations is considered. Wagner & Fisher (Wagner and Fisher, 1974) define a general notion of "distance" between two strings and present an algorithm for computing this distance. They state its relevance to spelling correction. If a cost is assigned to each edit operation, then the cost of a sequence of operations is the sum of the costs of each operation in the sequence. There may be a number of different sequences, each transforming A to B, each having different costs. The edit distance from string A to string B is defined as the cost of the sequence with minimum cost. They suggest that the cost functions could be set to depend on the particular characters affected by an edit operation: these could be used in spelling correction.

Following Wagner & Fisher, Backhouse (Backhouse, 1979) presents a mathematical model of 'error repair', where a repair is the sequence of operations used to transform one string into another: to correct an error. He includes transpositions in his edit operations, whereas Wagner & Fisher do not. He states that the philosophy behind his approach is to try to model the way the programmer would correct his own syntax errors. He also considers, in chapter 5 of his book (Backhouse, 1979), the repair of spelling errors, regular languages and context-free languages. His approach is "to consider all possible ways of repairing an input string and to define in each case the concept of a 'best' repair".

The best repair is a sequence of edit operations that will transform one string to another at minimum cost. The transformations that may be made between two strings (i.e. the edit operations that can be applied) are represented as a graph. If each operation is assigned a cost, the 'least-cost path' through the graph can be found, and will be the best repair, or the minimum cost repair. Detail of this algorithm is given in chapter 7. It appears then that there are methods that may be used for the application of multiple operations, despite Pollock's misgivings.

### 3.5. Summary

Computer programs that detect and correct spelling errors are available. They can be interactive or they may provide automatic detection and correction of errors.

Detection and correction of words misspelt as other words is difficult: research on the use of syntax and semantics has not progressed far enough to provide a solution to this problem. Detection methods currently used include digram and trigram analysis and dictionary look-up. The latter is likely to be of most use here.

Dictionary size (or length of word token list) affects speed and efficiency of error detection and correction. Larger dictionaries are more likely to include the intended word, but are also more likely to provide incorrect matches to misspellings or to provide other words as corrections. Shorter dictionaries are usually quicker to search but are less likely to contain the intended word.

Words in the dictionary may be represented as strings of letters (including abbreviations) or can be coded in other ways (for instance, hash coding). For the purposes of error detection, a 'word' must be defined, e.g. digits, apostrophes, hyphens and other special characters may or may not be seen as part of a word. Instead of representing complete words in a dictionary, root words and their affixes could be stored. This would permit a larger vocabulary to be represented in a smaller dictionary but requires reliable morphological analysis and reconstruction of words.

Methods of error correction include: looking at past misspellings; using measures of string similarity; editing rules; and string-to-string repair. Some approaches to spelling error correction use combinations of more than one of these methods. Of the methods, string-to-string repair is of greatest potential interest. The relative usefulness of these correction methods, in the context of their use with children with spelling disabilities, is discussed in chapter 5.

## Chapter 4

### Educational and technical constraints of the correction programs

#### 4.1. Introduction

In this chapter the theoretical conclusions drawn from the literature (chapter 2) are restated. The error classification scheme devised in this study is described. The two spelling correction programs developed in this study (referred to in chapter 1) are related to this classification scheme. These programs are described in outline in chapter 5 and in detail in chapter 7. The design of a computer program, incorporating the spelling correctors, is described in chapter 5. The educational and technical requirements of such a program are set out in this chapter. The failure of existing computer spelling correction programs in relation to these requirements is discussed (with reference to chapter 3).

#### 4.2. Theory

The most efficient method of spelling is by direct recall. Highly over-learned and familiar words are produced in response to meaning (Simon and Simon, 1973). There is direct access from the semantic information to the visual code (in whatever form this code takes); that is, a direct route from the semantic identity to the visuo-orthographic identity of the word. An indirect phonological route may also be used, in parallel (Baron, 1977). The word is output<sup>1</sup> from the visual code.

This means of retrieval will fail when:

- \* the speller is unsure about the spelling of part of a word;

---

<sup>1</sup>The role of visuo-motor co-ordination is not considered here.

- \* the word to be spelled is a homophone;
- \* the speller is unsure about the spelling of a whole word;
- \* the speller does not know the word.

Frith's three stages of the spelling process are accepted as the means by which words are spelt when direct recall fails:

1. there is analysis of the speech sounds of the word to derive appropriate phonemes;
2. phonemes are converted into graphemes by means of appropriate conversion rules or by analogy;
3. conventionally correct graphemes are selected from the phonetically plausible graphemes.

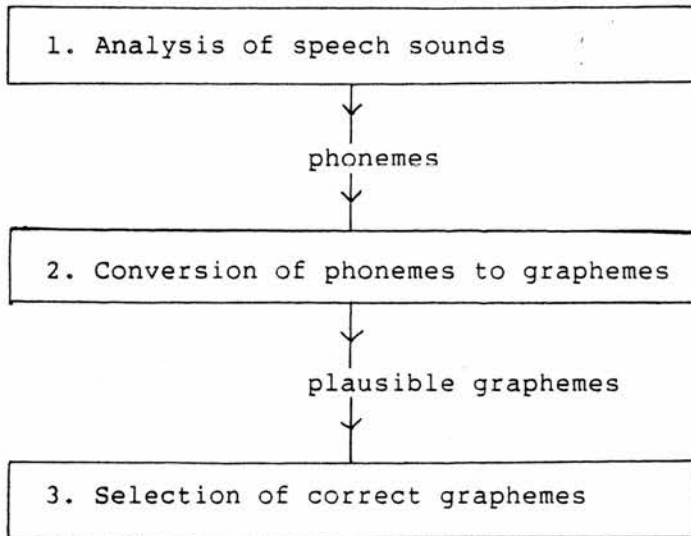
In the second and third stages of the process, the speller has to resort to using:

- \* phoneme-grapheme correspondences to generate plausible spellings;
- \* rules of morphology, syntax, permitted digraphs/trigraphs in the language, known irregularities, stress patterns, analogy, etc. to select the likely spelling.

In cases where the word is known, but only a partial or incomplete image can be recalled, selection of the most plausible graphemes may be facilitated by recognition of the word. Generated spellings are compared with some abstract graphemic representation of the word in the internal lexicon. This representation permits recognition of the word but is not sufficiently specified to enable production. If the word is unknown, i.e. it had not been seen before, there is no recognition information available to be used in the third stage of the process: no test can be made of the spelling generated.

Failure may occur at each of the three stages in the process. Difficulties can be experienced in segmenting the word and in the selection of the appropriate phonemes. Implausible graphemes may be selected to represent these phonemes; the phoneme-grapheme correspondences used by the child to generate graphemes from the phonemes may be incorrect or incomplete. Even when plausible graphemes have been selected, there are errors in choosing the

correct ones from these. The child's knowledge of orthographic structure may be inadequate: rules relating to word structure may be unknown or inappropriate in the particular context. This does not necessarily imply, however, that there is no regularity in the child's spelling: rules and correspondences can be regular, but incomplete or incorrect. Misspellings can be described in terms of 'bugs' in the rules or regular correspondences. The occurrence of these bugs can then be explained in terms of failure at one of these three stages in the spelling process, these three stages being:



### **4.3. Development of the error classification scheme and its relation to the correction programs**

The errors made by children with spelling difficulties, have been observed. A classification scheme has been devised to describe the regularities in these errors. The population, on whose errors the classification is based, is described in this section. The classification scheme is also presented, and its relation to the spelling correction programs given.

#### **4.3.1. The children studied**

A case study approach was taken in this research. The children studied were not compared or matched with any other group. The focus was on specific difficulties in spelling: some of the children studied were competent readers; most were not. The particular population defined were those children attending a Reading Unit (the 'South Bridge' Reading Unit) in Edinburgh.

Three different groups of children were involved in work carried out by the author. The first was a group of twelve children observed in the Reading Unit between January and March 1979 (all male); the second was seven children (one female) who took part in pilot study 1 (see chapter 6), attending the unit in May/June 1981; and the third was eight children (one female) involved in the second pilot study, in June 1983.

These children were predominantly boys aged nine to twelve years. They had usually had remedial teaching in their own schools, which had not been successful. The teacher will have noticed that they had problems, but may not have been able to deduce their difficulties accurately nor provide the help they needed. These boys were of approximately average intelligence; in some cases assessed as high IQ. Whilst they may have demonstrated below average ability in spelling (and often in reading), they had shown average (or above average) ability in other areas, e.g. mathematics. Their difficulties could not be directly attributed to low intelligence. An educational psychologist had assessed some of the children, but the Reading Unit teacher did not have full records of their assessment.

In Edinburgh, these children are required to attend a 'Reading Unit' for several

sessions a week. In the Reading Unit they receive intensive teaching from a trained remedial teacher. The aim is to improve their reading and writing skills, and for them to achieve some standard considered satisfactory for their age, before going on to secondary school.

A sample of some of the children's scores, on reading and spelling tests<sup>2</sup> is given in figure 4-1.

These tests were administered by the Reading Unit teacher. The children are those who were attending the Unit, and were observed by the author, in March 1979. Three examples of the writing produced by a different group of children, attending the Unit in 1981, are given in figure 1-1. The first group of children were observed, at work in the Reading Unit, over a period of one term. Errors made when the children were tested on words learnt from set lists (taken from "Aids to Spelling" books 1 and 2: Chandler) were recorded: the children were given these tests routinely, and had recorded all their answers in individual notebooks. Based on these errors, a classification of spelling errors was devised by the investigator and used to classify the errors made by the twelve children (Pain, 1980). The classification scheme described below is based on later revisions of this scheme.

#### 4.3.2. The error classification scheme

Errors made when spelling words may be described at different levels. At the lowest level of description, the misspelling can be described in terms of format:

- \* What letters are omitted?
- \* What letters are added?
- \* What letters are changed?
- \* What letters are transposed?

Information relating to the letters that make up the word is used: the letter positions, the particular characters, and the letter sequences. The relationship

---

<sup>2</sup>The scores of 20 and 21 were achieved by children who scored too low on the test to get an age grade

Child	Chronological age (years.months)	Spelling age (Burt tests)	Reading age (Schonell,R3)	Reading age
CR	11.0	7.7	7.7	7.6
	11.8		8.2	
FO	10.10	7.6	10.9	9.4
	11.7		10.6	
GR	10.2	8.4	10.8	9.2
	11.0		11.8	
KE	11.1	7.5	10.7	11.4
	11.3			9.8
MC	10.2	7.5	9.4	
	11.2		10.4	
RA	10.2	20	6.4	6.11
	10.10	21	7.4	7.2
	11.5		8.1	
SM	10.0	19	6.5	7.2
	10.8	6.7	7.1	7.11
	11.3		8.1	
WA	10.4	21	7.1	8.2
	11.1	7.3	7.7	8.2
	11.7		8.11	
AL	11.1	7.9	9.10	9.9
	12.0	8.0	11.4	
	12.7		10.7	

**Figure 4-1:** Sample of population: chronological ages, spelling ages, and reading ages

between the misspelling and the word can be defined in terms of alterations that would need to be made to the misspelling to correct it. These alterations – inserting, deleting, changing or transposing letters – will be referred to as editing operations, or edit operations. At the format level, therefore, misspellings are described in terms of sequences of edit operations required to match the misspelling to the intended word. For example, if 'dog' is spelt as 'bog', then the format error would be "change b to d". Similarly if 'chain' were

spelt as 'chane', then the format level of description would be "delete i" and "insert e". This level of analysis corresponds to that of Spache (Spache, 1940).

At the next level, errors can be considered in terms of rules: what are the general rules being used by the child to produce the misspellings? These rules may be correct, but applied in the wrong context; or they may be incomplete, or incorrect. By considering a number of similar format errors, general classes of errors can be described, and the rules or guides that the child is using suggested. Whilst the programs described in this thesis do not do this explicitly, how they could be extended to do so is considered in chapter 9. The description, therefore, includes the general class of character involved in the error, and the rule being applied or misapplied. A set of general classes of characters is given in figure 4-2.

The following misspellings can be described at the format level, or at the general level:

1a. mudle for muddle	2a. kit for kite
1b. aple for apple	2b. sak for sake
1c. puf for puff	2c. typ for type

At the format level, the first set of misspellings would be described by 'insert d', 'insert p' and 'insert f', respectively. The general class of character involved is "doubled consonant". The rule describing the error is "doubled consonant replaced by single consonant". In the second set of misspellings, the format error is the same for all three; 'insert e'. The general class of character involved is the "final 'magic e'". The rule is "represent the long vowel sound by vowel alone, omitting the modifying e".

At the third level of description the error is defined in terms of whether it is phonetic or non-phonetic. A phonetic error is described here as:

"an error which, if read aloud according to regular grapheme-phoneme conventions or by analogy, is indistinguishable from its correction"

The phoneme-grapheme correspondences that the child is using are considered: are correspondences from a generally accepted (legitimate) set being used? What are the actual correspondences being used? For example, the following are phonetic errors:

General class of character	Examples
Vowels	
- initial	a e i o u
- medial	a e i o u
- digraphs	ai ee aa ea ay
- diphthongs	ou oi oy
- final e	e
- modified by r	a e i o u
- modified by w or l	a e i o u
- final	a e i o u y
Consonants	
- initial	t b n s c g
- doubled	bb dd ff ck
- initial digraphs	ch sh th wh qu
- initial blends	st sc sm sw br cr tr pl fl gl
- c/g/s	ce ci cy ge gi gy s
Silent letters	
- b	bt mb
- g	gn
- h	rh ho ah hi wh gh
- gh	igh aigh ough
- k	kn
- l	lk ld lm
- p	ps pn
- t	st ft
- w	wr

**Figure 4-2:** General classes of characters

frunt	front	poot	put
thro	throw	chane	chain

In 'frunt' the /<sup>^</sup>/ is represented by 'u', as it is in 'sun'. The /EU/ in 'throw' is represented by 'o' in the misspelling, as it is in 'go'. Legitimate correspondences are also used in 'poot': 'oo' for /U/; and 'chane': 'a\_e' for /el/.

The description at these three levels can be related to the explanation of failure in terms of stages (see page 70). Failures occurring at the third stage in the process, in the selection of the correct grapheme from plausible ones, will be phonetic: plausible phoneme-grapheme correspondences have been used to

generate it. Rules will have been used in selecting the graphemes, but may have been used out of context; or the rule itself may be incorrect. The problem may be occurring at the level of visuo-orthographic information and can be described by the format of the error, initially. If the error is not a phonetic one, it could be due to a failure at the selection of plausible graphemes stage (the second stage in the process), or it could be at the first stage – the selection of the correct phoneme. If the description of the errors in terms of rules appears to be regular and consistent, but the errors made are not phonetic, the failure might be attributed to incorrect selection of phonemes. Alternatively, irregularity in the spelling itself may cause failure at the third stage, but may appear as phonetic errors.

Misspellings, therefore, can be described at three levels:

1. in terms of the format of the errors: the characters that need to be deleted, inserted, changed or transposed to match the misspelling to the correction;
2. in terms of general classes of characters involved and rules that are being applied or misapplied;
3. in terms of whether the misspelling is phonetic or non-phonetic, and the phoneme-grapheme correspondences being used.

It is suggested that there is a set of regularities that describe the English language at the visuo-orthographic level, and that 'correct' rules and guides are based on these regularities. These rules and guides relate to letter sequence, position, morphology, etc. An individual child will use rules based on a subset of these regularities, and may also have additional rules of his own. It is also suggested that the phonological regularities in English can be represented, to a large extent, by a phoneme-grapheme grammar. The child will use this grammar, or some variation of this grammar, in spelling. The grammars for individual children are not presented in this thesis.

#### 4.3.3. Relating the spelling correction programs to the classification scheme

The regularities used by the children when spelling perform a similar function to buggy procedures in arithmetic: where regularities exist that do not conform to 'correct rules' or generally accepted phoneme-grapheme correspondences, or where the spelling used would be appropriate in other contexts, we can say that there is a 'bug' in the child's spelling. These bugs can be described in terms of rules relating to visuo-orthographic structure, as in classification of errors at the general level, or in terms of phoneme-grapheme correspondences.

Brown and Burton (Brown and VanLehn, 1980) identify and describe buggy procedures in subtraction skills and use these to simulate the child's behaviour. The answers produced by the buggy procedures are compared with those produced by the children: that is, the errors are matched for a number of set problems. In the work described here, regularities in children's spelling were noted, including those regularities that conformed to normal spelling precedents and others that did not. These regularities, once noted, can be used to 'debug' the children's error: given the error the intended word can be inferred. For example, if there is a frequent confusion between 'b' and 'd' (that is, d is regularly replaced by b) and we have a word that is unrecognizable when spelt with a 'b', but makes sense when spelt with a 'd', we might infer that the intended word is that spelt with a 'd'. By noting the common b/d confusion we provide the information to debug the error.

Similarly, if we know that vowel sounds such as /i/ are often spelt 'ee', and a misspelling using 'ee' is encountered, we might look for a correction that has some other grapheme that can also represent /i/ in the same position as the 'ee' in the error, and infer that this was the intended word. By identifying phoneme-grapheme correspondences we can debug the errors.

The assumption is made here that sufficient regularities can be identified in the child's spelling to enable errors to be debugged: if there are not sufficient regularities there will be a failure in the debugging (or correction) of the child's errors. To test this assumption, it must be demonstrated that there is sufficient regularity for the errors to be corrected: that, given the errors, corrections can be inferred.

Two programs, 'editcost' and 'phoncode' were developed, incorporating both information about visuo-orthographic and phoneme-grapheme regularities in spelling, and frequent 'bugs' in these regularities.

The editcost program is based on an analysis of errors, described in terms of 'edit operations'. Errors made by the children were analysed at the format level and in terms of general classes of characters. Some information relating to the position and sequencing of characters was also incorporated e.g. in words where the initial phoneme is /f/, and is spelt 'ph', the 'ph' is often replaced by 'f'. Using information about the frequency of such bugs, errors are compared with potential corrections and matches selected. Having noted that 'f' for 'ph' in the initial position in a word is a frequent bug, a misspelling of 'phone' as 'fone' can be debugged, i.e. 'fone' can be corrected to 'phone'.

The phoncode program uses a phoneme-grapheme grammar. This grammar includes those correspondences used legitimately in spelling. It includes correspondences that are only used infrequently in specific contexts: if used more generally, in other contexts, these may appear as 'bugs' in spelling. 'Bugs' may also include 'regular' phoneme-grapheme correspondences used in words that are spelt irregularly. The grammar suggests plausible phonemes that any one grapheme (in the error, for example) was intended to represent. If, as in the case of the 'ee' -> /i/ example above, the grapheme used was plausible (e.g. 'beed' for 'bead') the intended phoneme can be identified and the misspelling debugged to identify likely corrections. The intended phoneme sequence is reconstructed from the actual (incorrect) graphemes. The errors corrected by this program correspond to those identified as 'phonetic' in the classification scheme. Detail of both programs is given in chapter 7.

Relating these to Frith's three stage model, the child has identified phonemes, generated graphemes and selected from these in attempting to spell a word. He is not certain, however, that he has produced the correct spelling: he may be sure that he has not, but be unable to produce the correction. If a correction could be offered, based on debugging his misspelling, the child's problem is changed from one of production to that of recognition (the latter being the easier task). The test of the success of debugging is whether a misspelling is corrected: there must be sufficient regularity in the spelling to permit this.

#### 4.4. Program Design Constraints: Educational

An important factor here for the child is that he is able to use his own productions, and is not restricted to reproducing lists of words. By basing his writing on project work and discussion, (of ideas that he may have, and of details such as the words he may wish to use), and providing him with a printed copy at the end of a session, we believe that he will be motivated to work on topics that interest him. He will learn to spell the words that he wishes to use, in context. The presentation and spelling of words out of context provides little reinforcement of the link between semantic, phonological and visuo-orthographic identities of a word.

Each child should have access to a dictionary containing those words most frequently used by children of his own age. This common dictionary might be based on word frequency counts of essays produced by children with no spelling disability. In addition to 'common words' in each child's dictionary, there will also be those words which he uses most often, (including proper names) or new words which he wishes to use relating to the current topic of his writing.

Discussion of possible topics (e.g. 'a haunted house', 'myself'), will stimulate ideas for compositions, and will also allow the teacher to predict those words that a child is likely to wish to use (which can then be added to the dictionary).

It is important that writing be seen as a developing process, that the child is encouraged to draft and edit his text. It is also desirable that he should be able to get a clean copy of his text once it is completed. A text editor could be provided to allow him to draft and edit his text. This editor must be easy to use: the child must be able and willing to use it. This would also permit the production of text without being concerned about handwriting.

When the child is writing it is important that he has some means of checking his spelling. The children of interest, however, are not able to check the spelling of words by looking them up in a standard dictionary: to look a word up in a dictionary requires having a reasonable idea of how to spell it. A spelling correction program could be provided that will enable him to check his spelling as he needs and if the spelling is not correct the correction should be

provided. Each child would have a basic common dictionary. This would be extended to include other words that he frequently used.

The computer does not replace the teacher, but provides facilities in addition to the skills of the teacher. The computer may also be used to perform tasks that the teacher cannot do (such as providing immediate feedback to a child). It is important that the child is active in learning, and is permitted to take initiative and responsibility for his actions. He must also be provided with feedback from his actions to enable him to modify what he is learning. In considering the teaching of children with specific learning difficulties, Malmquist (Malmquist, 1973) recommends that the child should be able to work at his own pace, be able to evaluate his own progress, and be motivated to learn.

The child should, therefore, be provided with supportive tools, of which he has control. He should also be provided with a suitable environment in which to use these tools. A more specific example of the way in which the child would work well will be given in chapter 5.

A working environment would be set up for the child, to include text editing facilities and interactive spelling checking tools. Work would be project based. There would be some discussion with the teacher, of the general content of the writing to be produced in each session. The child would then use the text editor to draft and alter his piece of text. Interactive spelling correction programs would enable him to check the spelling of any word, as he writes. An on-line dictionary will be accessible. The spelling correctors would, effectively, bootstrap his spelling competence and allow him to produce text largely free of spelling errors. Text could be produced, corrected, re-drafted, and finally printed for public appraisal.

#### 4.5. Program design constraints: technical

The spelling error correction program has to be able to consider errors in the form described in the error classification scheme: at format level; at the level of general classes of characters and rules; and at the phonetic level. It must be capable of correcting the misspellings produced by the children in the population of interest. The program must be able to take misspellings such as:

qorie      nedils      dacc      apone

and recognise them as:

quarry      needles      back      upon

Further examples of misspellings are given in chapter 1.

Damerau (Damerau, 1964) indicates that 80% of spelling errors are the result of one of the following four kinds of errors:

1. one wrong letter
2. one extra letter
3. one missing letter
4. two transposed letters.

These will be described here as 'single-error' misspellings. As can be seen in the examples above, some misspellings may involve more than one error of the type described by Damerau. These will be described as 'multiple-error' misspellings. Data collected for the children attending the Reading Unit indicates that a substantial proportion of misspellings involve multiple errors. The spelling correction program must, therefore, be able to correct multiple error misspellings.

An adult of average spelling ability may be capable of correcting an error, once it has been detected, without the aid of a spelling correction program. Thus, when using a correction program, it will not present too great a problem if the correction is not always offered. Accuracy of the spelling correction algorithm may be permitted to suffer for the sake of speed. To reduce the search space, the assumption may be made that the first letter of a misspelt word is correct. Pollock and Zamora (Pollock and Zamora, 1984), indicate that, for 92.2% of words misspelt, this assumption is correct. For the remaining 7.8%, it is

assumed the user can correct the error himself. For the Reading Unit children, more difficulty is encountered with 'self-correction'. It is important that the intended word is not rejected at an early stage, even if this means a larger search space is needed. Additionally, first letter errors are more frequent in this group than in the general population. A spelling corrector for this group cannot be designed on the assumption that the first letter of any misspelling is correct.

As well as ensuring that the word required is included in the words offered as corrections for a misspelling, it is also necessary to restrict the number of possible corrections offered. If a large number of words are offered as corrections, the child will have difficulty selecting the correct one. However if a very small number are offered there is a greater risk that the desired word will be excluded.

Children with spelling difficulties make a larger number of errors in short words than the general population. Some correction programs provide no correction of short word misspellings: these are needed, however, in the programs to be used here.

If the misspellings to be corrected are those made in response to testing of a pre-specified list of words, then the target (correct) spelling is already known, and correction is straightforward. If the child is not to be constrained by spelling words lists, however, but is to produce his own composition, then the correction program must have knowledge of his vocabulary. It must also be able to predict, from a misspelling, the intended word from that vocabulary. At the same time, it is desirable that the vocabulary can be easily extended.

If the correction program is to be used by the child, to check and correct misspellings, then the program must be interactive. Whilst it might also have the facility to take a completed piece of text, and suggest corrections for misspellings in it, this should not be the main mode of use. If the child is having difficulty with the spelling of a particular word, he should be able to request immediate feedback from the program, before continuing his writing.

Information about the relationship between the misspelling and intended word is

to be used to carry out the spelling correction. As described earlier in this chapter, this information will be in terms of edit operations, general classes of characters and rules, and phoneme-grapheme correspondences. It would be desirable to use the information used by the correction program to debug or correct the error i.e. to describe the bug more explicitly. For example, if phoneme-grapheme correspondences are used to correct the error, we might also wish to use this information to state exactly what correspondences are being used by the particular child. We would like to be able to extract the specific information used to correct any single error. Whilst this information is not currently extracted in either the editcost or phoncode programs, it is desirable to leave open the option of doing so at a later stage.

#### 4.6. A summary of the overall requirements

In summary, the program to be used should fulfil the following educational constraints:

1. it should not be based on fixed word lists; the vocabulary should be extensible;
2. the correction of errors made in free writing of English should be permitted; this includes text that may not be perfectly "grammatical";
3. the correction facility should be interactive and provide immediate feedback whenever requested;
4. it should permit the child to generate and alter text with ease;
5. the individual should be able to use the program as a tool and be in control of it.

The technical requirements may be restated as:

1. being able to deal with multiple errors;
2. making no assumption of "first letter correct";
3. having the facility to correct short word errors;
4. retaining sufficient information for errors to be reconstructed;
5. providing information about phonetic equivalence of words;
6. providing the correction, assuming it is in the dictionary, and not rejecting it at any stage in the process of selection of candidates;
7. providing some reasonable means of pruning the list of candidate corrections to a number from which the user might select the correction, i.e. not too many "final corrections" are presented.

#### 4.7. Failure of existing programs to satisfy the requirements

The spelling correction programs described in chapter 3 are considered in relation to the requirements stated above.

Considering the pattern-matching methods of error correction described, they have each been applied with varying degrees of success. It is difficult to compare their performances because of their different applications, dictionary sizes, etc.

Of interest here is their likely success within the domain chosen, i.e. for use with children with learning difficulties in spelling.

The programs using the edit operation algorithm (Damerau; Dec-10; Durham, Lamb & Saxe; Mor & Frankel; Peterson; Omond; Pollock & Zamora), all assume single error misspellings. The exception to this is Backhouse's work (Backhouse, 1979): his algorithm could be adapted for incorporation in a correction program, though currently only assigns unit cost to all operations. The Dec-10 program also assumes first letter correct, so would omit these errors from its candidates list (Peterson, 1980a). Durham, Lamb & Saxe state that:

"the spelling correction algorithm is quite adequate for our requirements [user-interface applications] though it clearly is not optimal for the general correction application in prose." (Durham et al, 1983), p.767

No method of pruning the candidate list is provided by Peterson, Omond or Mor & Frankel. Pollock & Zamora indicate that short (3 and 4 character) letter misspellings present problems, though the similarity keys that they use might be of practical use in finding candidates for corrections. Most of the programs could be used interactively. None of them deals with phonetic information about errors.

Of the programs using matching substring algorithms (Galli and Yamada, 1968; Morgan, 1970), the problem is the criteria of 'matching' required. The Spelltell program requires the input part of the word to be correct: this cannot be guaranteed. It also provides no means of pruning the list of matches found.

Solo's matching algorithm requires a percentage match for some part of the strings. Correction will fail in short word misspellings and in cases where words are misspelt with different (but perhaps phonetically similar) letters: e.g. 'phone' misspelt as 'fon' would only provide 2 matches and would be rejected. Baskin & Selfridge's algorithms (Alberga, 1967) also rely on percentage of string matched, and would be subject to the same criticism. In order to accept this type of error the criteria of acceptance would have to be so low that a great many other 'candidates' would also be generated. These algorithms do not exclude the possibility of correcting multiple error misspellings or of correcting misspellings where the first letter is incorrect. However, they do not provide any means of using phonetic information in correction. Representation of errors as abbreviations (Blair, 1960; Davidson, 1962) requires that at least 4 letters in the spelling should be correct. Davidson requires that the first letter be correct. Alberga (Alberga, 1967) claims that abbreviation methods of spelling correction 'fail badly', (p. 311.)

Again, these methods present difficulties for short misspellings and for pruning the candidate list without rejecting the correction at an early stage. In addition they do not provide phonetic information.

'Measures of similarity' algorithms could generally be useful for providing a 'shortlist of candidates' but do not necessarily provide information to prune this list (Faulk; Morrison; Pollock, 1982). The Interlisp program is designed to be used for Lisp programming (Teitelman, 1978), and not for English text. The Plato correction program (Tenczar and Golden, 1962) appears to be suitable in many respects:

- \* dealing with children's text;
- \* allowing multiple errors;
- \* being interactive;
- \* offering matches based on a number of features (phonetic features might be included);
- \* allowing a small number of corrections to be offered (by altering 'criteria for bit matches');
- \* permitting short word errors;

\* and claims of good performance.

However, it presents three major problems:

1. it is difficult to retrieve the information used to make the correction  
- the error cannot be reconstructed;
2. some reliance is placed on the first letter being correct (though this might be altered);
3. it is a commercial program and the coding details are not available to be modified, and presents too large a task to be "devised from scratch".

Returning to the issue of using syntactic and semantic information in spelling correction, there is current research in both areas, by Fass (semantics) and Atwell (syntax) (Fass, 1984; Atwell, 1983). Neither was working in these areas at the time that research on this thesis commenced. In both cases work is 'in progress' and not yet completed. Mitton, specifically working on the problem of context in relation to human spelling error correction, only recently commenced work on the problem (Mitton, 1984a).

Munnecke (Munnecke, 1980), using the Soundex code, attempts to correct misspelt names using phonetic information. This coding, however, relies on the first letter being correct. It also provides all matches to the coded name, and no means of pruning these. In some cases it would not code the error in a way that matched the correction.

The program that seems most suited to solving the problem of 'phonetic equivalence' of words is that proposed by Savin in Alberga, 1967. According to Alberga, in Savin's approach:

" ... the pronunciations of each word in the stored dictionary is encoded and associated with the word. A set of rules for the pronunciation of the language involved, in this case English, is then used to determine if the input string could be a phonetic representation of any of the words in the list. Of necessity, the rules must be somewhat context-free, as the type of error which one wishes to detect may very well be the use of some letter or letter group in the wrong context due to faulty analogy." (Alberga, 1967, p.304)

Alberga states that he implements these ideas, as a function, "Phone". Phone

tests a misspelling to determine if it could be a representation of the pronunciation of the correctly spelled word.

"... A list of rules was compiled from a variety of sources [Pronunciation guides and English grammar textbooks]; the rules consist of the various representations which occur in English for each of the sounds. These rules were grouped into three sets, those in which one or more letters represent a single sound, those in which a letter or group of letters represent two sounds, and those in which a letter or group of letters is silent ... The program searches for a representation of each sound or pair of sounds which, when placed together, with silent letters between where necessary, matches the misspelling." (Alberga, 1967, p.308)

Alberga gives further no details of the rules used, and no reference for them except the pronunciation and English guides from which they were taken. In evaluating the function, he concludes that the method was 'clearly unusable' and 'failed rather badly', (p.311). It is difficult to see exactly how he got these results, as insufficient detail is provided to permit replication. No reference can be found in the literature, to any work by Savin on this problem, by the author or by Fass (1983).

There is research in the literature that could be used as a basis for construction of a "phonetic error correction program". Ellovitz, Johnson, McHugh and Shore (Ellovitz et al, 1976) describe work on letter-to-sound rules for automatic translation of English text to phonetics. Ciarcia (Ciarcia, 1982) describes the 'Microvox text-to-speech synthesizer'. Both papers deal, however, with rules for correct translation from letters to sounds, or from text to speech: of interest here are 'misapplied' conversion rules, and 'mispronunciations'. The work of Hanna et al. (Hanna et al, 1966) (see chapter 2) is also of relevance here. Whilst no program currently exists to provide information about the phonetic equivalence of spelling errors, there appears to be information which may be used to enable its design, (Yannakoudakis and Fawthrop, 1983b; Venezky, 1966).

A spelling correction program that might have been suitable, is that developed by Yannakoudakis & Fawthrop (Yannakoudakis and Fawthrop, 1983b). This program was built in parallel with that described in this thesis, and so was not available for consideration at the time of development. As it is of particular relevance, it will be described below.

Their corrector is based on an analysis of spelling errors described in a previous paper (mentioned in chapter 2) (Yannakoudakis and Fawthrop, 1983a). In this they showed that, when comparing 'error form' and 'dictionary form', spelling errors follow specific patterns and rules, and also that search for matches can be restricted to certain parts of the dictionary. Their spelling correction system,

"contains information of possible pronunciations of the word and also of the nature of spelling and typing errors as made by adults."  
p.101.

It uses no knowledge of context or semantic structure. It is designed for word processing applications. A word is looked for in the dictionary which contains 93,769 words, and if it is not found it is passed to the algorithm for correction.

A small part of the dictionary is searched word by word. If one or two differences are found between the error form and the dictionary form, and the differences follow any of the rules, the dictionary word may indicate a 'correction' for the error, and is added to a list of choices. Bayesian statistics are used to select the most likely word from the choice list. In developing the algorithm, feedback was provided by repeated correction of a list of errors by adult 'bad spellers': all subjective Bayesian probabilities were varied, in turn, and set to the value giving the maximum number of corrections.

The program was tested on 4 sets of data:

1. 'typical' list of errors made by adults;
2. list of errors made by adults classing themselves as 'bad spellers';
3. list of spelling errors made by 12th grade children, taken from Masters (Masters, 1927);
4. data taken from Damerau (infrequent misspellings of words taken from news stories).

Assuming the word was in the dictionary, the success rate for each of the above lists (that is, the % of times the correction was in the 4 - or fewer - words offered) was:

list 1. 94% ; list 2. 86% ; list 3. 84% ; list 4. 82% ;

Over all four lists, 90% of errors were corrected. In 95% of these, the

correction was the only word offered. It may be noted that the results for the 'bad spellers' and children are not as good as for adults.

Considering the 'requirements for a program', this program would satisfy most of them. However, it is based on frequency of errors made by adults, some of whom are 'bad spellers', and so could still be better 'tuned' to the children of interest. The assumption that these children make errors that can be described by 'rules', and that this assumption can be tested by implementation of 'rules' in a correction program, still has to be tested. Yannakoudakis and Fawthrop make the same assumptions for adults, and test them successfully. Their approach and methodology is of relevance here, though the errors will be different: it is not clear that the rules, and error likelihood functions, would apply directly to the children studied in this thesis. Additionally, Yannakoudakis & Fawthrop make the assumption of only single errors in short words and this may not apply.

None of the programs reviewed above fully satisfies the requirements for spelling correction and use with children with learning difficulties in spelling. Therefore, in chapter 5 the design of a computer program that will do so is given. The detail of this program is given in chapter 7.

## Chapter 5

### Design and use of a computer program to incorporate the editcost and phoncode correctors

#### 5.1. Introduction

The editcost and phoncode spelling correctors mentioned in chapter 4 have been developed: their design and use is outlined in this chapter. They are described in detail in chapter 7. The editcost corrector has been incorporated in an interactive spelling checking program, together with facilities for adding additional words to the editcost dictionary (= 'addword') and for dictionary reference (= 'lookup'). This program was used by a group of children from the Reading Unit, as described in the second study (study 2), chapter 6. Whilst the phoncode program has not been used directly by the children, it has been tested on the full corpus of their errors (see chapter 8).

Text editors for use by children exist: one such editor ('Walter') is described in chapter 6. Assumptions about its use by children from the Reading Unit were tested in the first study (study 1) described in the same chapter. This text editor was designed and implemented by Sharples (Sharples, 1984) and used by him in a study of children's creative writing skills. It was chosen for use in study 1 for the following reasons:

1. children in the Sharples study had not had difficulty using it;
2. it fulfilled the necessary requirements of allowing the child to interactively create and alter text;
3. it was available for use on a machine that the children could access easily.

It is not suggested that Walter was the 'ideal text editor' however: other more suitable editors may exist or could be developed, e.g. screen editors with multiple windows, perhaps. These will not be considered further here.

In this chapter, a description is given of how both the spelling correctors and other facilities could be incorporated in a larger program<sup>1</sup>, satisfying the educational and technical design constraints imposed in chapter 4. A text editor, based on that used by Sharples, forms the shell of this program. The means by which these constraints are satisfied are summarised at the end of this chapter. The content and structure of the dictionaries used in study 2 are described, together with an example of the dictionary. A brief discussion of error detection and correction follows, together with short descriptions of the editcost and phoncode correction programs and other existing facilities.

## 5.2. Example of a hypothetical session

A hypothetical session is described in this section, based on an actual session taken from the second study. This example is constructed from:

- \* a protocol collected in the second study, where the editcost program, addword and lookup facilities were used;
- \* data from the first study where the text editor, Walter, was used;

The phoncode correction program is also incorporated in this example session.

During any one session, the child's writing is based on a particular topic. The topic for each session will have been decided at the end of the preceding session. It will relate to a project that the child will work on over a number of sessions. The stimulus for writing will take one of a number of forms: it might be an interview, a demonstration, 'horror story' swapping, or 'last night's football match'. The teacher/investigator will discuss with the children what they might write.

The conventions used below for distinguishing between text displayed by the computer and text input by the user (the child) are:

an example of text displayed by the computer  
*an example of text input by the user*

---

<sup>1</sup>This larger program does not currently exist. Walter was written in Pop-2, whereas the spelling correctors and other programs were written in Pascal. Given a version of Walter in Pascal, however, it would not be difficult to implement this imagined program

The dictionaries relating to the chosen topic are set up before the start of the session.

```
%gospel/
type in filenames as requested - type 'no' to stop
name of file to be used
peterossdict
another file?
generaldict
another file?
no
dictionaries set up
```

Each child then goes to a terminal and 'logs-on':

```
Please type your first name and then press the RETURN button
Steven
Hello, Steven
```

If the teacher is aware of any words that the child might need, that are not already in his dictionary, she adds them to the dictionary for the session:

```
w: addword
What word do you want to add
: turtle
Give a meaning or example
: a small robot used in turtle geometry to draw shapes
w:
```

The child may then start to write his new composition, giving the appropriate command for adding text to the text editor:

```
w: new
story: logo is done by Peter Ross. <cr>
story: logo is about the turtel <cr>
story: <cr>
w:
```

w: is the command level prompt for the editor; story: is the text prompt; carriage return <cr> causes the cursor to move to the next line. A second <cr> immediately after the 'story:' prompt causes a return to command level. The story may then be continued, or edited, or a spelling checked.

w: *check*

What word do you want to check

: *turtel*

Wait a minute while I check it.

It could be

turtle

true

turned

terrible

Type the word that you want (or 'no' if it is not there)

: *turtle*

I will change it.

The checked word is changed to the correction, the story (so far) retyped on the screen, and the cursor placed at the end of the text. The child can continue with his story:

story: logo is done by Peter Ross.

story: logo is about the turtle *a little*<cr>

story: *bugy* <cr>

story: <cr>

w: *check*

What word do you want to check

: <cr>

Wait a minute while I check it.

It could be

buggy

boy

by

body

Type the word that you want (or 'no' if it is not there)

: *buggy*

I will change it.

In this case, if the child types <cr> when asked which word to check, the program assumes that the last word typed is to be checked. It is automatically replaced:

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy *which is controlled by a*<cr>  
 story: *commputer. it costs \$350* <cr>  
 story: <cr>

w: *check*

What word do you want to check

: *commputer*

Wait a minute while I check it.

It could be

computer

compute

computes

computing

Type the word that you want (or 'no' if it is not there)

: *computer*

I will change it.

Earlier words in the text may also be checked and replaced.

Words may also be changed without using the spelling checker:

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: *The turtle can make shapes.*  
 story: <cr>

w: *change*

old words: *make*

new words: *draw*

old words: <cr>

The first occurrence of 'old words' in the story is changed to whatever is put after 'new words'. More changes may be typed after the next 'old words' prompt. Alternatively <cr> returns the child to the story, with the changes made:

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: The turtle can draw shapes. /k<cr>  
 story: *can be in matham* <cr>  
 story: <cr>

w: *check*

What word do you want to check

: *mathamatics*

Wait a minute while I check it.

It could be

    mathematics

    materials

    material

    manipulate

Type the word that you want (or 'no' if it is not there)

: *mathematics*

In this case the checked word ('mathamatics') does not match to any word in the text (only 'matham' was typed). It cannot be changed automatically. The user is prompted:

Is it the last word that you want changed

: *yes*

If the user types 'yes' or 'y' the last word typed will be deleted and the word selected from the options offered will be substituted:

I will change it.

and the story is retyped:

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: The turtle can draw shapes. It  
 story: can be in mathematics

If the user types 'no' (or any response other than 'yes' or 'y') then he is given the choice of appending the option selected to the text (after the last typed word), or to delete some other word in the text and insert it in its place:

Do you want to add this word  
 : *no*  
 Do you want to change a word  
 : *yes*  
 Which word  
 : *matham*

If a successful match is made the word is changed. If no match is made, or if the response to 'change a word' was not yes, then no change is made. The story is retyped on the screen and the composition continues.

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: The turtle can draw shapes. It  
 story: can be in mathematics. *There is*  
 story: *another type of turtle for the BBC*  
 story: *which use fisher tecnic* <cr>  
 story: <cr>

w: *check*

What word do you want to check

: *tecnic*

Wait a minute while I check it.

It could be

technic  
 teaching  
 taking  
 think

Type the word that you want (or 'no' if it is not there)

: *techic*

That is not one of the words - try copying it again

: *technic*

I will change it.

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: The turtle can draw shapes. It  
 story: can be in mathematics. *There is*  
 story: *another type of turtle for the BBC*  
 story: *which use fisher technic*

If the correction is miscopied i.e. it does not match any of the options, the user is asked to retype it. If retyped correctly, the program proceeds. If there is still no match no change is made:

Sorry, I still cannot change it.

a message is printed and he is returned to the text.

Sometimes, when a word is checked, the required correction may be missing from those offered:

```
story: logo is done by Peter Ross.
story: logo is about the turtle a little
story: buggy which is controlled by a
story: computer. it costs $350.
story: The turtle can draw shapes. It
story: can be in mathematics. There is
story: another type of turtle for the BBC
story: which use fisher technic.
story: The govemement has bought <cr>
story: <cr>
```

w: *check*

What word do you want to check

: *bought*

Wait a minute while I check it.

It could be

brought

bough

both

but

Type the word that you want (or 'no' if it is not there)

: *no*

Is it one of these:

boat

The reason that the correction is not offered is either that it is not in the dictionary or that the editcost program fails to find it. Further options are offered by the phonetic coding algorithm. One of these may be accepted, in the same way that the previous options were accepted:

Is it one of these:

boat

: *yes*

Type the word that you want (or 'no' if it is not there)

: *boat*

I'll change it.

If the correct word is still not offered (as in this case, where the checked word is spelt correctly), the user types 'no' again and is asked to try respelling the word:

Type the word that you want (or 'no' if it is not there')

: *no*

Think hard about the spelling - type it again to check it  
(or type 'no' if you think that the word is missing)

:

If a word is retyped, then it is checked as before. If 'no' is typed the user is asked if he wants a word added:

: *no*

Do you want a word added to the dictionary

: *yes*

Ask your teacher to help you.

If a response other than 'no' or 'n' is typed the addword procedure is prompted as before. Otherwise, the user is returned to the text.

story: logo is done by Peter Ross.  
story: logo is about the turtle a little  
story: buggy which is controlled by a  
story: computer. it costs \$350.  
story: The turtle can draw shapes. It  
story: can be in mathematics. There is  
story: another type of turtle for the BBC  
story: which use fisher technic  
story: The govemement has bought *lots of*  
story: *turtles for schools. The turtle*  
story: *works by mottor* <cr>  
story: <cr>

w: *check*

What word do you want to check

: *mottor*

Wait a minute while I check it.

It could be

motor

mother

more

not

Type the word that you want (or 'no' if it is not there)

: *lookup*

Which word do you want to look up in the dictionary  
 : *motor*  
 motor  
 = a machine to make things move  
 Is this the word that you want  
 : *yes*  
 I'll change it.

The user may delay continuation of the correction, and look up the definition of a word in the dictionary, using the 'lookup' command. If the user types 'yes' when asked if this is the word he wanted, the program continues as if he had selected this word as the correct option. Otherwise, the response is taken as a 'no' to the 'type the word you want - no if it's not there'.

story: logo is done by Peter Ross.  
 story: logo is about the turtle a little  
 story: buggy which is controlled by a  
 story: computer. it costs \$350.  
 story: The turtle can draw shapes. It  
 story: can be in mathematics. There is  
 story: another type of turtle for the BBC  
 story: which use fisher technic  
 story: The govemement has bought lots of  
 story: turtles for schools. The turtle  
 story: works by motors *controlling wheels*.  
 story: *The B.B.C Buggy cost \$60.* <cr>  
 story: <cr>

w: *save*  
 name of file  
 : *logo*  
 save finished  
 w:

When the story is complete it can be saved in a named file in the user's area. It can be retrieved again with the command 'recall':

w: *recall*  
 name of file  
 : *logo*

The story will be printed on the display. The cursor will be positioned at the end of the text. More text may be added to the story. The text may be printed out (on a lineprinter) with the command 'print'.

At the end of the session the user types 'goodbye'. The user is prompted for any story that has not been saved since last editing:

w: *goodbye*

Do you wish to save your last story:

: *y*

name of file

: *turtle*

save finished

Goodbye, Steven

The following sections relate to the interactive program incorporating editcost, addwords, and lookup, and to the phoncode corrector, and also provide the basis for the program illustrated above.

### 5.3. Dictionaries

For each session each child works from a 'session dictionary' in his working area. It is possible to read any number of dictionary files into the session dictionary at the start of a session (=sessiondict). In general, the files used are:

1. a general dictionary file (=generaldict)
2. a dictionary of vocabulary for the particular topic of that session  
e.g. horrordict is a dictionary of 'horror story' vocabulary.

If, as a result of discussion of the topic, it is apparent that there are additional words that will be required, these can be added. It is possible to add further words at any later point in the session. These words are not, however automatically added to the permanent dictionary file, only to the temporary sessiondict. The addword facility could be altered to enable permanent storing of the added dictionary words. Provision for phonetic coding of the added words would have to be made.

#### 5.3.1. Size and content

Examples of some of the dictionary files used and their sizes are as follows:

File	Topic	Approx. No. of words
generaldict	frequently used words	700
myselfdict	physical description of person	171
footballdict	football match review	221
horrordict	horror stories	173
islandict	description of a desert island	239
turtledict	using the logo turtle	82
turtle2dict	further vocabulary for logo	250
perqdict	using the perq computer (used in conjunction with turtle2dict)	67
photodict	developing a film	105

Some dictionary files were specifically compiled for writing up interviews with members of the Artificial Intelligence department:

File	Topic	Approx. No. of words
davewysedict	technical drawing/ mechanical engineering	266
patamblerdict	robotics	230
peterossdict	using logo in education	185
mikeshdict	Open University distance learning	171

In some sessions two topic dictionaries might be used, either for related vocabulary, or in cases where the child is finishing one story and about to start another. On average, the dictionary used in any one session will contain between 750 and 1000 words.

In addition, a definition or example is provided for each word in the dictionary. See figure 5-1 for sections from the `generaldict` and `turtle2dict`.

### 5.3.2. Dictionary structure

Each word is represented in two dictionaries. In the pattern-matching ('editcost') dictionary each word is read in and stored as a string, with pointers both to a string representing a definition or example, and to the next word. Words are indexed by 'first character': a number of special first characters are defined (see chapter 7 for details).

In the phonetic coding dictionary, a file of words and codes for their phonetic representation is converted into a tree structure: each node is a phoneme and each daughter is the next phoneme in some word. Thus, words represented by three phonemes will match to nodes at three levels of the tree, the final node including a pointer to the string (word) that the phonemes represent (see chapter 7).

## 5.4. Definition of 'a word' and affixes

A working definition of a word is 'a sequence of alphabetic characters delimited by spaces, linefeeds or punctuation'.

These characters may be upper or lower case, 'a' to 'z'. Apostrophes, hyphens and digits in words are not regarded as characters. Contractions such as 'doesn't' and 'I'll' are not included in the dictionary. All input is converted to

Sections from turtle2dict

define  
 say what it means, "define a procedure for drawing a square"  
 dome  
 round cover, like the top half of a ball  
 drops  
 lets fall, "the goalie drops the ball and kicks it"  
 .....  
 letters  
 what words are made up of, "there are 5 letters in chair"  
 logo  
 a language used for computing  
 .....  
 turtle  
 a small robot, "you can draw shapes with the logo turtle"  
 create  
 make something that has never been made before  
 database  
 a collection of data or information in a computer  
 degrees  
 measure of amount of turning

Sections from the generaldict

different  
 not the same, "she wears different shoes each day"  
 discovered  
 found, "I discovered gold in the box"  
 .....  
 glad  
 pleased, happy, "I am glad that I brought my umbrella"  
 go  
 leave, "go away, go home"  
 goes  
 leaves, "he goes home every weekend"  
 gone  
 left, "all the people had gone and the place was deserted"  
 good  
 not bad; nice, fine, "it was a good film, I enjoyed it"  
 .....  
 new  
 not old, just made, "is that a new jacket, I have not seen it before"  
 news  
 information, "have you heard the news, we won"  
 next  
 one after, "and the next in the queue is me"  
 night  
 darkness, not day, "it was a dark and wet night"

**Figure 5-1:** Example sections from the dictionary

lower case for matching: all words in the dictionary are in lower case, and so

all 'options for corrections' are given in lower case. This includes corrections of proper names. No affix checking or stripping algorithm is used.

## **5.5. Error Detection**

No spelling error detection process was used in the interactive checking program in study 2 or in the example session above. The user offered the word to be checked: he had to decide for himself which words were possible misspellings. In 'checking a word' corrections from the dictionary are offered, including the input misspelling if it matches a dictionary word.

The program could be extended to include a facility to 'check a passage'. All words in the passage that are not found in the dictionary would be queried as misspellings. For example, in the text above, the word 'govement' is neither checked nor corrected. If the whole passage were checked, this 'word' (misspelling) would be highlighted in some way, to indicate to the user that it should be checked. Options for the correction might be offered automatically. In checking the passage, the definition of 'a word' would have to be made more precise. Digits appearing as part of a word could be queried, other digits in the text ignored. Punctuation characters „+:-?!“() spaces and linefeeds could be taken as delimiters. Words containing other non-alphabetic characters or apostrophes could be queried as errors.

## **5.6. Error Correction**

### **5.6.1. Syntax**

As was discussed in chapter 3, whilst it is desirable to use syntactic information in spelling error detection and correction there are many difficulties in doing so. Attempts were made to find solutions to these at an early stage in this project.

The first 'possible solution' considered was to take some existing natural language parsing program and adapt it. If regularities could be found in the children's grammar, and these conformed (to a large extent) to the grammar used in the parsing program, then the grammar in the parser could be extended to parse the children's text. Defining the grammar for the children's text,

however, proved impossible! It was also realised that, as a substantial proportion of the words would be misspelt, the parse would have to be capable of dealing with ill-formed input. This is not a trivial problem, and is in fact a separate area of natural language research (Fass and Wilks, 1983).

A different approach was considered: instead of attempting to parse a complete sentence, the immediate context alone could be considered. Given any word that is labelled as a part of speech, the likelihood of any other part of speech occurring adjacent to it can be calculated. Thus, given two adjacent words with ambiguous part of speech labels the most likely combination of labels for that position can be predicted. Additionally, if it could be assumed that a small number of frequently occurring, correctly spelt, words exist – for example a set of function words such as 'a, the, at, and, of, from, for...' – then these words (in conjunction with the likelihoods of adjacent labels) could be used to construct 'templates' for sequences of labels for parts of sentences. So, for any word not found in the dictionary a misspelling would be assumed. All function words found in the text would be assumed to be correct. Templates would be matched to word sequences. A 'correction' label would be assigned to all misspellings according to the prediction of its most likely label, taken from matching to the template. This correction label would then be used in selection of candidates for corrections, in addition to other information. Words misspelt as other words would be detected if their 'possible labels' did not match to the most likely label assigned for that position in the template. Again, this 'most likely label' might be used in correction.

However, the assumption that the set of function words will always be spelt correctly cannot be made. Additionally, a great deal of work would be required to calculate the likelihood of words occurring in relative positions: it would be a thesis project in itself to provide a type-labelling system for ill-formed and incorrect input. Consequently, work on this particular problem was not followed up in this thesis. Atwell (Atwell, 1983) is currently working on this problem.

### 5.6.2. Semantics

No work was carried out, in this thesis, on semantic analysis. The use of semantic information in spelling correction is the subject of Fass's 1983 MSc thesis (Fass, 1983), and his current PhD thesis (Fass, 1984) (see chapter 3).

There is semantic constraint placed on words considered as correction candidates, in that words not appearing in the topic related or general dictionaries are not offered i.e. words offered are mostly those relating to the topic of interest. Additionally, the facility for referencing the definitions of words ('lookup') provides some semantic information as an aid to the user.

## 5.7. The editcost program

The editcost program takes the word to be corrected (prompted after the command 'check' in the example) input by the user, and the sessiondict. Sections of the dictionary are selected for comparison with the input word (=inpw): these words form the shortlist. The selection is based on the initial letters of inpw and its length. The shortlisting function is described in section 7.2.2. Each word on the shortlist is compared with inpw. For each word, the cost of editing inpw to match it is calculated. The cost is dependent upon the edit operation and the particular letters involved. Detail of the calculation of editcosts is given in chapter 7. The four words from the dictionary with least editcost are offered to the user, in ascending order of cost, as options for the correct spelling of the input word (see the example in section 5.2).

To some extent the decision to offer four options is an arbitrary one. Certain constraints, however, do limit the number of words to be offered to the user. Firstly, it is not possible to guarantee that the closest match to the checked word will always be the required word: for the most bizarre spellings (taken out of context) only telepathy could guarantee accurate correction! Further information about unrecognizable words could be obtained from the context or perhaps by recognition of the spoken word. This is not possible here. Educationally, it is not wholly undesirable that the program should fail to correct these words: the child is left with some incentive to spell the word 'as close to correct as possible'.

Secondly, the greater the number of options offered, the more likely it is that the correction will be amongst them. The larger the number of words offered, the more difficult the task of selection will be for the child. With a greater number they will be more likely to become confused, or be unwilling to read until they find the correct one.

It is necessary to select the minimum number of words to be offered such that the correction is likely to be amongst them. In the initial development of the editcost algorithm (before actual use with the children) varying numbers of words were offered. Offering more than four words did not appear to greatly increase the frequency with which the correction was included in the options. The assumption is made that, if the word is in the dictionary and the checked word is a reasonable enough approximation<sup>2</sup> to it, the four closest matches should include it. Chapter 8 gives detail of the program's success in this.

### **5.8. The phoncode program**

The phonetic coding program takes the input word, *inpw*, and a dictionary and selects those words in the dictionary that might be considered phonetically equivalent to the *inpw*. Using a table of grapheme-phoneme correspondences, the *inpw* is split into all combinations of graphemes with their corresponding phonemes: it is effectively being parsed to generate all phoneme sequences, according to the grapheme-phoneme grammar. A sequence of phonemes, generated by the grammar, will be referred to as a 'phoneme sentence'. The dictionary is represented as a tree of phonemes. The *inpw* 'phoneme sentences' are matched to the phonemes in the tree. If a path in the tree matches a phoneme sentence, and that path represents a word, then the *inpw* is considered to be a possible phonetic equivalent to that word from the dictionary. All matches are found by exhaustive search and are offered to the user (the 'four option' condition applies only to the editcost program). See section 7.3 for more detail of the phoncode program.

---

<sup>2</sup>'Reasonable enough approximation' here means that the spelling could normally be recognised, by a competent speller, as the intended word.

## 5.9. Other facilities

### 5.9.1. The lookup facility

The lookup procedure takes a word from the user, prompted when the command 'lookup' is given, and accesses the editcost dictionary. It looks for the word in the dictionary (see figure 5-1) and prints out its stored definition.

*w:lookup*

Which word do you want to look up in the dictionary?

*w:true*

true

= not false, "is it true that you have a job"

*w:lookup*

Which word do you want to look up in the dictionary?

*w:stick*

stick

= fix; piece of wood, "stick that poster on the wall"

*w:lookup*

Which word do you want to look up in the dictionary?

*w:turtle*

turtle

= a small robot, "you can draw shapes with the logo turtle"

If the word to be looked up is incorrectly spelt, or if it is not in the dictionary, then a message will be printed:

*w:lookup*

Which word do you want to look up in the dictionary?

*w:stik*

stik

= is not in the dictionary

*w:lookup*

Which word do you want to look up in the dictionary?

*w:specifications*

specifications

= is not in the dictionary

Both 'lookup' and 'look up' are accepted as commands.

### 5.9.2. The addword facility

If, as a result of discussion of the writing topic, it is found that there are words that the child might wish to use but that are not in the dictionary, they can be added with the addword procedure. A definition is also added. Additionally, if the child finds that a word he wishes to check is not in the dictionary it may be added, using addword. The child is encouraged to ask the teacher/investigator to add the word, to prevent misspelt words being added to the dictionary. (Unless the teacher/investigator is able to phonetically code the word, it can only be added to the editcost program dictionary).

*w: add word*

What word do you want to add?

*specification*

Give a meaning or example

*how you specify something*

*w: addword*

What word do you want to add?

*possible*

Give a meaning or example

*can be done*

The procedure accepts both 'addword' and 'add word' as commands.

## 5.10. Relating the design to the requirements

The design of the program presented in this chapter will be related to the program requirements, given in chapter 4, section 4.6.

The correction programs are not restricted to fixed word lists or predefined dictations, though they are limited by the topic and general dictionaries forming the session dictionary. The addword facility permits further extension, however. The child can use the corrector to check any word he wishes whilst writing. It is used interactively. A word can be corrected at any point in the writing process: the child has control of the tool.

If the correctors were incorporated in a text editor then the child would be able to generate and alter text with ease, assuming he can use the text editor. This assumption is tested in study 1, described in chapter 6.

Neither the editcost nor the phoncode programs are restricted to dealing with single error misspellings. There is no assumption of "first letter correct", though there is some restriction on which alternatives are permitted in the editcost program. Both programs are able to correct short word errors.

In the editcost program, the sequence of edit operations used to transform the misspelling into the correction is stored. It could easily be used to reconstruct the error. Similarly, grapheme-phoneme correspondences used by the phoncode program can be recorded and the error reproduced. This could provide information about the phoneme-grapheme correspondences being used by the child. It can also be used in classifying pairs of words as 'phonetically equivalent'.

The programs' success in providing the correction, assuming it is in the dictionary, is considered in chapter 8. It was considered important that the correction should not be rejected at any stage in the process of selection of candidates.

The editcost program permits any number of candidate corrections to be offered to the user, although only four are currently offered, whilst the phoncode program only gives (a small number of) exact matches: therefore, not too many "final corrections" will be offered to the user.

A number of questions relating to the programs need considering: questions relating to assumptions about the way in which the children will use the programs are addressed in chapter 6. In chapter 8 the performance of the correction programs is assessed.

## Chapter 6

### Testing assumptions about the use of the program

#### 6.1. Introduction

In chapter 5 a number of assumptions were made about the way in which the program described would be used. These assumptions were tested in two studies, described in this chapter. Children from the Reading Unit took part in these studies.

There were three main purposes of the first study:

1. to test whether children with spelling disabilities would be both able and willing to use a simple text editor to write stories;
2. to examine the ways in which the text editor was used by the children, and to consider ways it might be extended;
3. to provide samples of the children's errors, made in free writing on pre-specified topics.

In the second study, the editcost spelling corrector was used to test whether or not:

1. a child would use the spelling corrector to check the spelling of words;
2. he would be able to select the required word from those presented by the correction program;
3. he would recognize when the required word was not presented by the correction program.

## 6.2. Study 1

### 6.2.1. Introduction

The dyslexic child is not usually highly motivated to write using pencil and paper. He is also unwilling to proof-read his work; to look for errors, check them and correct them. It is argued, in chapter 4, that the provision of computer based tools, such as text editors, would increase his motivation to write. He would use simple text editing commands to draft his composition, to make changes to it, and to correct the errors. The final copy that he produces would not reveal errors made in earlier attempts: it would be a piece of work that he would not feel the need to conceal. Additionally, by not constraining the subject matter and vocabulary that the child uses in his writing, he will be more interested in, and see more purpose to, his production.

The child must be able to cope with the commands needed to use the text editor, and to use the keyboard to input his text and commands. Typing itself may create problems: different skills are involved in typing and handwriting. In typing, any spelling information available through the kinetic sense is lost: the child is not concerned with 'making the shape of the letter' but with recognising it and pressing the appropriate key on the keyboard. 'Typing errors' may be introduced, but these are more likely to occur with typists typing at speed. 'Handwriting errors' will be unimportant. Additional confusion may be caused by the fact that the keyboard is labelled in upper case: the child is more familiar with lower case. The added feature of the delete key may outweigh the difficulties of using an eraser.

Text editors have been used successfully by children who have no specific spelling difficulties (Sharples, 1984). Evidence is needed to demonstrate that children with these difficulties can also use them. The argument that the child will be able and motivated to write, using the computer, has to be supported: if evidence cannot be provided then the tool, as designed, would be of no use.

Information gathered through the observation of the children in this study influenced the design of the program described in chapter 5. It should be noted that no spelling correction was provided for use in this study. The way in which

the correction programs might be incorporated was considered. The child was encouraged to guess the spelling of words initially and to ask for help with correcting errors he spotted when proof-reading. In effect, the investigator performed the function of the 'spelling corrector' here.

All the spelling errors made by children in this study were recorded. The corpus of errors is given in appendix C. These errors were used in the design and implementation of the spelling correction programs (see chapter 7).

### 6.2.2. Method

#### **Subjects**

The children attending the Reading Unit usually did so for one or more sessions a week, each session lasting approximately one and a half hours. Those normally attending during two particular sessions were selected to take part in the study. There were four children in the first group (group 1), and three in the second group (group 2). Those children in group 2 (J.M., G.Q., and M.W. - all boys) were considered by the Reading Unit teacher to be moderately able (when compared with all children attending the Unit). Three in group 1 (S.S. - the only girl, N.M., and C.M.) were considered by the teacher to be the least able ("hopeless cases?"). The fourth child in group 1 (L.B.) was thought to be very bright. Exact details of age, I.Q. scores, and reading test results were not provided.

All the children had difficulties with writing, and some also with reading (although their reading had generally improved whilst attending the Unit). Specific problems were mentioned in the cases of two children: S.S. - dysphasia, and C.M. - auditory perceptual difficulties. Several children also had behavioural problems (G.Q., M.W., and L.B.).

The aim of this study was to observe how well the children coped with the system, and to collect data on the children's errors: it was not to assess improvement in the children's abilities, and so no control group was used.

#### **Apparatus**

A simple text editing program, called Walter (Word ALTERer), was used by the

children. This program, written in POP-2, was developed by Sharples, and has been used by him to investigate children's use of written language (Sharples, 1984). Walter allowed each child to write new stories, to edit them, and to store and retrieve them. A summary of the commands available is given in appendix A.

The program was run on the Artificial Intelligence (University of Edinburgh) Department's PDP-11/60 computer under UNIX. Each child had his own user number and copy of the program. Each used a keyboard V.D.U. to type text into the computer and a printer provided final copies of the stories. Everything which appeared on the V.D.U. screen during a session was recorded. (These records were not available for the first session, and occasionally lost in later sessions due to technical malfunctions).

Worksheets were used in the first few sessions, providing instructions for the use of Walter. A summary sheet of commands was used for most later sessions.

### **Procedure**

Each group of children attended for six sessions. Group 2 started two weeks before group 1, so the project ran for eight weeks in total. All children attended all sessions, with the exception of S.S who missed two sessions. The actual time spent using the computer, for each child, was 45-55 minutes per session. The children were collected from the Unit and brought to the A.I. Department, and then returned to the unit after the session, (hence the reduced time spent using the computer).

The first part of each session was spent discussing the topic on which the stories were to be based. The first session was concerned with "How to use Walter". Later sessions involved either preselected topics (e.g. turtle logo, an adventure) or topics arising from general conversation on the way to the A.I. Department (e.g. horror stories).

After some discussion of a topic, the group were taken through to the terminal room, and were logged onto the machine by the investigator. During the first session worksheets were used to familiarise the children with Walter. In later

sessions they wrote stories based on the topics discussed at the start of the session. They typed each story straight into the machine. They were encouraged to write whatever they wished, and to attempt to spell words correctly, but not to worry unduly if they were unsure of a spelling - to guess it if they didn't know it. When the story was complete (that is, when they were sure that they had finished it, or at the end of the allocated time), they saved it in a "memory" file. At this point they were asked by the investigator to indicate those words in their story that they thought that they might have misspelt. (The amount of prompting at this stage varied). Each child was then prompted to correct those words misspelt. When all spelling errors (and some of the grammatical errors) had been corrected the story was saved and then printed out. Each child received a copy of all of his stories, corrected and neatly printed.

Notes were made after each session (based upon observation and recordings) of how each child coped with the keyboard and with Walter. Each child's spelling errors and problems arising during the session - with the program, or with the machine - were also recorded.

Features such as the amount of help each child was given with spellings when first writing a story, and how much he was prompted when using different commands, were varied. Different children required varying amounts of help. Consideration of the most effective ways of using the program was influenced by the results of these variations.

The children's attitudes were assessed informally throughout the project. There were asked how they felt about the project whilst going to and from the department, and during the last session. Some feedback on the children's attitudes was also obtained from the Reading Unit teacher.

### 6.2.3. Results

Over the six sessions, five short stories were written by N.M., three by S.S. and four by each of the other children. Copies of the corrected versions of the stories are provided (in appendix A). Lengths of the stories varied from twenty words (N.M., "Peter (my cousin)"), to ninety-eight words (G.Q., "Blackhand"). (The story "Scream" by J.M. was one hundred and ninety-four words, but was

actually written away from the terminal). Information about the length of their pencil and paper stories was not provided.

The children were caused some frustration initially by the problems in logging on, machine crashes, and by being thrown out of Walter when they leaned on the keys whilst thinking! These problems were resolved in later sessions.

It had previously been suggested that making the children type their stories might cause them to make errors. They were very slow typists, and tended to hover over the keyboard searching for a letter. Because they were slow, however, they did not make many actual typing errors (and were able to correct those that they did make). There were also no problems caused by using the upper case keyboard (whilst seeing lower case on the v.d.u.), and the 'qwerty' keyboard presented no special difficulties. The children learned to use the special function keys easily, (e.g. 'return', 'delete', 'shift', and 'space'), although they occasionally forgot to press the 'return' key at the end of a command line, and caused problems when using 'back-space' instead of 'delete'.

The Walter editor commands were all used by most children. After the first session, little difficulty was encountered with them. The main problem was that of typing a filename instead of a command after the prompt e.g. "W: train" instead of "W: recall train". There was also some misspelling of commands.

In general, all the children in the two groups got on well with the system. The children in group 1 worked particularly well together, often offering suggestions and advice to each other. This was the less able of the two groups, but once they had got used to the system, they were able to use the editing commands. They had more difficulty in judging whether words that they had written were correct or not, and in correcting them (with the exception of LB), but all of them appeared to be well motivated. The children in group 2 were generally better at using commands, wrote longer stories, were better at correcting errors, but were more easily distracted, and not as enthusiastic and hard-working as those in group 1. More detail is given about individuals in appendix A.

The children said they enjoyed using the computer to write stories, and appeared to do so. Some (especially in group 2) were less willing to go back

and read over their work and make changes, whilst others were very keen to correct their errors. All, however, persevered to produce a 'good error-free copy' of their compositions. They also requested extra printed copies, on a number of occasions, to show to friends and family. One child commented that he preferred writing with the computer to writing on paper, because if he made an error it was easy to correct it without it showing with the computer, whereas if you had to rub it out there was always a hole in the paper! Group 1 were especially excited about using the computer, and children in both groups were keen to come back to do further work. The Reading Unit teacher sat in on several occasions: she was pleased with the work produced and happy for the children to take part in later studies.

It was found that the children generally coped well with using the text editor. There were some problems: the 'change' command changed all occurrences of a string in the text: if only one specific occurrence was to be changed the string had to be put in context; the Walter program decapitalised all first words in the 'change' function, causing some confusion when changing capitalised words - this was solved by including the preceding (lower case) word in the change; each new piece of text had to be added as a separate 'new' story, and could not be added to an existing story; the original story might be saved, then changes made and a copy printed, but no copy of the corrected version saved.

The results obtained from the collection of errors made by the children are summarised and discussed in chapter 7, figures 7-5 and 7-6.

#### 6.2.4. Discussion

The children learned to cope with problems with the editor, as any user learning to use an unfamiliar text editor would have to. A different text editor might have been more suitable for their use, or Walter altered. Over a long period of use the children would get more practice with the keyboard and editor. A screen editor might, however, have alleviated some of these problems.

The children were asked to guess the spelling of words, and to ask the investigator for help when correcting them. It was sometimes difficult to resist giving the child the correct spelling initially, without encouraging him to guess it. It was also difficult to deal with all the children who wanted help with

correction at any one time. It usually took two sessions to complete a story – one to write it and the second to proof-read and correct it. This meant that sometimes a child forgot what a word was meant to be; for example, one child had used the word 'fapyer' in one session but had forgotten by the next session what it was meant to be – it was never corrected! The incorporation of the spelling corrector, as a 'check' function, would solve these problems. The child would be able to exit from writing text temporarily and ask the program to 'check' the spelling of a word, and would be given the correction (or possible corrections). He would get immediate help, and not have to wait until the teacher/investigator was free. He would then be able to write his story, checking the spellings he was uncertain of as he went, and could also use the facility when proof-reading. The children expressed an interest in having such a facility.

From the results it can be seen that the children were able to write using the computer. They had a large amount of control over what they wrote, and their errors were not seen after they had been corrected. They were able to produce a corrected, printed copy of their stories, which they could show to other people. Their writing had a function, in communicating their ideas, and was not just 'an exercise in a book'. No assessment was attempted of improvements in their spelling ability as a result of interaction with the computer, however.

### 6.3. Study 2

#### 6.3.1. Introduction

In order to use the spelling checking facility described in chapter 5, the child types in his spelling of the word he wishes to check, and then selects the required word from those options offered. Evidence from the first study supports the need for this on-line checking facility. The child cannot ask the teacher for help with the spelling of every word, and it would be difficult for him to check the spelling using a conventional dictionary.

It is necessary to test whether or not the child would be able to select the required word from those offered by the checking program. If the child is unable to recognize the intended word when presented with it, the checking facility would be of no use. If the program were always to offer the correct word, and no others, then this would not need to be tested. It is not possible to guarantee that the closest match to the checked word will be the required word, however. If the word is in the dictionary, and the checked word is a reasonable enough approximation to it, then the four closest matches should include it (section 8 gives detail of the program's success in this). 'Reasonable enough approximation' here means that the spelling could normally be recognized, by a competent speller, as the intended word. It is not desirable for him to select the wrong word from those being offered: this might reinforce learning of a misspelling. If bizarre and unrecognizable words (relative to the children's usual spellings) are produced, then it is not wholly desirable that the program should correct them: there would be incentive for the spelling to be 'as close to correct as possible'.

It was hypothesized in chapter 1 that the child would be able to recognize correct spellings even if he could not produce them. This hypothesis is further discussed in chapter 4. It was tested in this study.

Study 2 tested whether the child would use the spelling correction facility, and also examined the ways in which it was used. The text editor was not incorporated, however, nor was specific phonetic information about words used (see chapter 5) to select options for corrections. The editcost program, plus lookup and addword facilities was used.

The reason for use of the editcost spelling corrector, but not the phoncode corrector, was that it was desirable to test whether the checking program could be used at a stage when the phoncode program was not fully developed. The result was that fewer options were provided for the misspellings: this was, therefore, a more stringent test.

This study was designed to answer a number of specific questions:

1. If the child is unsure about the spelling of a word does he use the computer spelling program to check it, or does he guess the spelling?
2. If he checks the spelling, is he able to select the required word from those offered by the program in cases where the word is present?
3. If the required word is not offered when checked, does the child realise this?
4. More generally, in what ways is the program used?

Information obtained in this study was also used in evaluating the performance of the editcost program (reference chapter 8).

### 6.3.2. Method

#### **Subjects**

The children who took part in this study were selected on the basis of their availability at the time of the study.

The first group consisted of four children: D.R., F.R., D.V., and T.E., all boys aged between nine and eleven years. They were considered to be of average ability by the Reading Unit teacher (in relation to other children attending the Unit).

In the second group there were four children: three boys, D.S., S.T., and G.R., and a girl, M.A., all aged between eleven and twelve years. The Reading Unit teacher considered the children in this group to be exceptionally bright (reported IQ's : G.R. > 128, others > 135).

All the children had difficulty with spelling, and some also experienced some difficulty in reading.

## **Apparatus**

The children used a version of the editcost spelling checking program. This program enabled them to check the spelling of any one word by comparing it with words stored in an on-line dictionary and displaying the four closest matches to the child's word/misspelling. This checking algorithm is discussed fully in chapter 7. The main example used in chapter 7 is based on a session in which one of the children was using the checker. The on-line dictionary differed for each child, for each session (see section 5.3.1). Each dictionary consisted of a file of commonly used words, including those words used frequently by the particular child, and a "topic vocabulary" based on the words likely to be needed for the particular session's task. For a large number of the words, definitions and examples of their use were added to the dictionary. This information was either provided automatically or on request.

The version of the spelling checking program used in this study was written in PASCAL and was run on the Artificial Intelligence Department's VAX 11/750 under UNIX. Each child had his own copy of the program and used a keyboard v.d.u. Everything that appeared on the v.d.u. screen during a session was recorded. Notebooks were provided for the children to write their stories in.

## **Procedure**

Group 1 attended for eleven sessions, over a period of 3 weeks. Of the four children D.R. was absent for 5 sessions and F.R. was absent for session 11. T.E. and D.V. attended all sessions. Approximately forty-five minutes per session were spent at the computer terminal.

Group 2 attended three (weekly) sessions, each of one and a half hours. All four children attended all three sessions.

Each group focussed on a particular project during the study. Group 1 focussed on producing articles for a magazine: topics included a sports review, a horror story, a visit to the departmental workshop and an interview. In the first session they were shown how to use the "check" program (the interactive version of the editcost program), whilst writing a description of themselves. Group 2's project was a report on the Department of Artificial Intelligence. They

interviewed members of staff and were given demonstrations of computers and programs. They were shown how to use the check program, whilst writing up their interviews, in the first session.

At the start of each session some stimulation for writing was provided: e.g. each child carried out an interview, or saw a computer demonstration, or there was a group discussion focussing on a particular task. Once they had decided what they would be writing about, each child sat next to a computer terminal and started writing. The spelling checking program was running on the computer.

They wrote in pencil in an exercise book. They were told that whenever they were unsure of the spelling of a word they were to try to write it anyway, and then to use the checking program to check it. The program would offer them four possibilities for the word that they wanted. If they thought that the word they needed was among these four they were to cross through their written attempt (if it was not the same) and write in the correct word. They were requested not to rub out or scribble over any words. If the word was correct (the same as the selected option) they were to leave it as it was. It was emphasized to them that this was to be their working draft, in the exercise book, and that it did not matter if it was a bit messy as it was to be typed later anyway by the investigator.

If they thought that the word they needed was not present in the four offered by the checking program, they were told that they should look at their spelling again. If they were not certain it was correct, they were to check it again with a different spelling. This will be referred to as 're-checking' the spelling. However, if they believed that it was correct but not in the dictionary they should tell the investigator who would then add it to the dictionary. In these cases, the word was added immediately without the child seeing it, and the child checked his original misspelling again. When they had completed the piece of writing they were asked to proof-read it and to check any words that they were still unsure of.

Varied instructions were given regarding the use of the lookup facility. For the first week (group 1, sessions 1 to 4; group 2, session 1) the children were not

told about the facility, as the dictionary definitions were not complete. For the following sessions, group 1 were told that they could "lookup" a word either to see if it was actually in the dictionary or to find out its definition. It was suggested to them that if they thought that one of the options offered by the checker was the correct one, but were not certain, they could "lookup" the word to confirm that it was the required word (or otherwise). In addition, TE and FR were automatically given a one-line definition or example with each word offered by the checker. In group 2, DI and GR were told the same as DR and DV in group 1, and were not provided with definitions automatically. MA and ST were told that they could 'lookup' a word to see if it was in the dictionary, but that definitions were not provided for all words.

Everything that appeared on each child's v.d.u. screen was recorded for later analysis. The investigator was present in each session and also observed the children and made notes on these observations.

Printed copies of each corrected piece of writing were produced by the investigator: if errors still remained in these the child was prompted to check and correct them (either by using the program or by self-correction).

### 6.3.3. Results

For each child, from his writing in his exercise book and from the recordings of the v.d.u screen<sup>1</sup> each session, words that had been altered and those that were incorrectly spelt were noted. Additionally, a small number of words that appeared in the dribble files (having been used with check or lookup) but had not been written were noted.

Each of these altered and checked words was recorded according to whether:

1. It was initially
  - a. correct - left as correct;
  - b. correct - self-altered to an error;

---

<sup>1</sup>These will be referred to as dribble files.

- c. correct - self-altered to another correctly spelt word;
- d. incorrect - left as an error;
- e. incorrect - self-altered to another correctly spelt word;
- f. incorrect - self-altered to a different error;

2. It was

- a. self-corrected by the child
- b. checked using the program

If the word was checked (2.b) then it was noted whether:

- 1. The correct word was offered, noting its position in the four options.
- 2. The correct word was not offered.

If the word was offered, whether the word originally written was:

- 1. left correct;
- 2. altered to be correct;
- 3. left as an error;
- 4. altered to an error;
- 5. or some other course of action was taken.

If the correct word was not offered the course of action that the child took was noted, including:

- \* asking the investigator to add it to the dictionary and checking it again;
- \* rechecking it with a different spelling;
- \* leaving the word as it was;
- \* changing the word to one of those offered;
- \* using the lookup facility.

Specific questions that were asked are now considered.

### **Checking the spelling**

If the child is unsure about the spelling of a word, does he

1. use the computer spelling program to check it?
2. guess the spelling?

The number of cases in which the child used the checking program to check a word were counted. These included a number of words that were correctly spelt, but that the child checked using the spelling checking program. The words in this category, therefore, could have been words that were correct or incorrect initially, and that were corrected, left as errors or changed, after checking.

The number of errors left unchecked or unchanged, plus the number of self-corrected or changed words, were classified as guessing the spelling. There were a small number of words that were left incorrect because the child had no time left to read over and check them, by himself or with the program: these are also included in this category.

In total, 395 words were checked using the program including 99 that were initially correct. 173 words were left as errors, including 25 errors that the child had no time to check. Figures for individual children are given in figure 6-1.

Considering these results it can be seen that the spelling checker was used in a large number of cases to check spellings. It is also of interest to note that of those 296 incorrectly spelt words (395 total less 99 initially correct) checked using the computer program, 216 were corrected; of the 148 errors self-checked (173 total less 25 unchecked) only 55 were corrected.

### **Selecting the required word**

Could the child select the required word from the four offered by the checking program in cases where the word was present?

The following were counted:

1. the number of times that the child selected the correct word i.e. left the spelling unchanged if it was correct, or corrected it if it was not;

	Number of words checked using the program		Number of words left as errors or self-altered	
	Total	(initially correct)	Total (no time left to check)	
Group 1				
TE	88	(18)	35	(11)
DV	53	(8)	13	(0)
FR	91	(18)	32	(10)
DR	52	(11)	18	(0)
Group 2				
DI	18	(9)	19	(0)
ST	29	(14)	12	(0)
MA	37	(17)	9	(4)
GR	27	(8)	35	(0)
Totals	395	(99)	173	(25)

**Figure 6-1:** Words checked using the correction program

2. the number of times that he did not select the correct word i.e. left an error uncorrected or altered the word to be a different word or an error.

The former category will be taken as evidence that the child could select the correct word and the latter as that the child could not. Some words did not fall clearly into either category e.g. if the lookup facility was used after checking a word it may be that the child thinks he recognizes the word as correct (or not, if he is "looking up" the wrong word!) but is still not certain. These words are not counted here. Individual figures are given in figure 6-2.

<sup>2</sup> The figures for TE and FR are not included in the totals in this figure. This is because they were automatically given the "lookup" information.

146 words were checked and counted. For 128 of them there was evidence that the correct word had been selected when offered. In only 18 cases was the correct word not selected. So, in 87.7% of cases the correct word was selected when offered.

---

<sup>2</sup>The total number of words checked and offered in figure 6-2 plus those checked and not offered in figure 6-3 does not equal those checked in figure 6-1 because if a word is not offered, but is then added and re-checked (and is then offered) it may occur in both figure 6-2 and figure 6-3.

	Total offered correctly selected	Number of correct words selected	Number of correct words not selected	Percentage selected when offered
Group 1				
DV	44	40	4	90.9%
DR	37	32	5	86.5%
Group 2				
DI	10	7	3	70%
ST	12	11	1	91.7%
MA	24	21	3	87.5%
GR	19	17	2	89.5%
Totals:	146	128	18	87.7%
( TE	62	53	9	85.5%)
( FR	78	73	5	93.6%)

**Figure 6-2:** Words selected when offered by the correction program

### **Words not offered by the checking program**

If the required word was not offered when checked did the child realise that it was not presented in the four options?

The following were counted: the number of times that the child re-checked a word, or left an error to stand, or left a correct spelling unchanged, or asked for help, was taken to indicate that the child did realise that the word was not presented.

The number of times that the child selected one of the presented words, and altered the original word, was taken as indicating that the child did not recognize that the required word had not been presented.

The results show that of 115 cases in which the required word was not offered, in only 13 (c.11%) of these did the child not realise that the word had not been presented. Therefore, in 89% of cases the child did recognize that the required word had not been offered. Individual figures are given in figure 6-3.

### **Other ways the program was used**

	Total	Child realises word not presented	Child does not realise word not presented	Percentage child does not realise word not presented
Group 1				
TE	26	20	6	23.1%
DV	11	9	2	18.2%
FR	28	28	0	0%
DR	13	10	3	23.1%
Group 2				
DI	5	5	0	0%
ST	11	11	0	0%
MA	11	10	1	9.1%
GR	10	9	1	10%
Totals:	115	102	13	11.3%

**Figure 6-3:** Child realises the correct word is not presented

In general the program was used as intended. However, there were some exceptions to this. DI on a number of occasions used the 'lookup' facility to check the spelling of a word: if he spelt the word correctly and it was in the dictionary, the word and definition were printed. The lookup facility was faster than checking the spelling of the word, and he took advantage of this. For cases where the desired word was not in the dictionary, ST added it himself on several occasions. In some cases he spelt it correctly. In others he did not. When it was spelt incorrectly, he would then check his spelling, as before, and his incorrect spelling would be confirmed. The children were not encouraged to use the addword facility.

#### 6.3.4. Discussion

The evidence from this study shows that the children were able, and willing, to use the spelling checking facility provided. They checked the spelling of a substantial proportion of the words they were unsure about although in some cases the lookup facility was used for effectively the same purpose. One of the main problems was the slowness of the program.

In 87.7% of cases where the correct word was offered by the editcost program,

it was selected by the child as the intended word. There were a number of occasions, however, when the correction was not offered. The program's performance is assessed in chapter 8.

In cases where the intended word was not offered by the correction program, the child did realise that it was not present: in some cases where the children were unsure they re-checked the word, or looked up the definition of one of those offered. In only 11% of cases where the word was not offered did the child not appear to realise this.

The children in group 1 used the program over a greater number of sessions, and therefore checked more words. Individual children varied in their ability to select the correct word, or to detect when the correct word was not offered.

The observation that the children would be able to recognize the correct spelling, even if they cannot produce it, was made. This observation needs to be confirmed by further, more rigorous, testing.

#### **6.4. Summary of results from the two studies**

The results from each of the two studies will be summarised briefly.

- \* Children with learning difficulties in spelling are both able and willing to use a simple text editor to write stories.
- \* The text editor could be usefully extended to incorporate an error checking and detecting facility.
- \* If a spelling correction program is provided, it is used to check the spellings that the child is uncertain of.
- \* If the required word is offered by the program the child is generally able to recognize it. If it is not offered, he usually realises it is not there.

The assumptions made about the use of the program, therefore, were generally supported by these two studies. In addition, evidence was presented suggesting that, whilst the children might not be able to produce the correct spelling of a word, they were able to recognize it.

## Chapter 7

### Detail of the editcost and phoncode programs

#### 7.1. Introduction

In this chapter the editcost and phoncode spelling correction programs are described in detail.

#### 7.2. Calculating the minimum cost repair: the editcost program

##### 7.2.1. General overview

A shortlist of dictionary words is selected from the session dictionary as possible candidates for the misspelling. If all words in the dictionary were considered then a very large number of comparisons using the costing algorithm would have to be made. The object of shortlisting is to reduce this number whilst retaining the correction of the misspelling in the shortlist. The selection depends upon the first two characters of the misspelling and its length.

The misspelling (termed the 'inpw') is compared with each dictionary word (termed the 'dictw') on the shortlist. The cost of editing the inpw to match the dictw is calculated (see section 7.2.5). Those dictws with lowest edit cost are saved. The four dictws with lowest cost are offered to the user as options for the correction of the misspelling (the inpw).

##### 7.2.2. Shortlisting candidates from the dictionary: 'shortlist'

From the misspellings made by the observed group (see chapter 4) and the group in study 1 (see chapter 6), the following were noted:

- \* the initial two letters of the word and misspelling and the frequency of each, in cases where the first letter of a word was misspelt;

- \* the range of differences in length between misspellings and corrections.

### **Initial Characters**

There were 'first letter' errors found in approximately 8% of misspellings made by children in the study 1. For these cases, therefore, it could not be assumed that the first letter was spelt correctly. First letter confusions found were used to construct a table of 'alternative first letters'. For any inpw only those words from the dictionary with the same initial letter, or an alternative found from the table, are considered as candidate misspellings.

In some cases the 'alternative first letter' is extended to consider the first two characters of the inpw and dictw. In doing so the alternatives for the inpw are made more specific and the shortlist further reduced (though the risk of omission of the correction is increased). For example, if the misspelling is 'rite' for 'write' then all the words in the dictionary with initial letters 'r' or 'wr' would be included in the shortlist whilst those others beginning with 'w' would not.

The dictionary is indexed by 'first character', where this may actually be indicated by the initial two characters of the word, or by the first character only. In cases where two letters are used they are represented by a single character in the range A-Z. These 'special cases' of first character are shown in figure 7-1

The first two characters of the inpw are read. If they match a special case the first characters are then represented by the appropriate letter; otherwise they are represented by the first letter of inpw. Alternatives for the first character are found by table look-up. They are given in figure 7-1. All words in the section of the dictionary indexed by the first character, and by the alternatives for it, are included in the shortlist (subject to length constraints). For example, if the inpw were 'foto' the first character would be 'f' and the alternatives for it would be P(=ph), T(=th) and v; so all words in the dictionary with initial letters 'ph', 'th', 'v' and 'f' would be considered for the shortlist of candidates.

### **Length constraints**

Any dictionary word considered to be too long or too short to be the correction

initial letter(s)	represented by	alternatives
a	a	auoei
b	b	bdp
c	c	ckgsqCK
ch	C	Ccjsk
d	d	db
e	e	ea iy
f	f	fPTv
g	g	gcjG
gn	G	GKgn
h	h	hHwO
ho	H	Hho
i	i	iuea
j	j	jCg
k	k	kKcq
kn	K	KGnkc
l	l	lL
m	m	mn
n	n	nGKNm
o	o	oaHwO
p	p	pbPN
pn	N	Npn
ph	P	Ppfv
ps	S	Sps
q	q	qkc
r	r	rR
s	s	sSc
t	t	tT
th	T	TfvtP
u	u	uoy
v	v	vTf
w	w	wWO
wo	O	Oow
wr	R	Rwr
wh	W	WwhOR
x	x	xe
y	y	yui
z	z	zs

**Figure 7-1:** First character alternatives for shortlisting

of the inpw is omitted from the alternative words on the shortlist. The range of length permitted was determined by comparison of misspelling and corrections lengths for the observed group and study 1 group. The dictw is shortlisted if:

- \* the difference between the length of the *inpw* (=inl) and the length of the *dictw* (=dwl) is less than 4, when both inl and dwl are less than 10, or
- \* the difference between inl and dwl is less than or equal to  $(dwl/3)+1$ , when inl or dwl is 10 or greater.

Those candidates on the shortlist, to be passed to the costing algorithm are, therefore, those words from the session dictionary satisfying two conditions:

1. with the same initial letter, or with alternative initial letters, as that of the input word;
2. with length within the range specified above.

### 7.2.3. The editcost algorithm

At the format level of classification, errors are described in terms of the editing operation that is applied to the misspelling to get the correct word (see chapter 4). For example, to correct 'kat' to 'cat', 'k' must be changed to 'c': the error is reversed to produce the correction by the application of the edit operation 'change' to 'k' to produce 'c'.

The basic editing correction method is used in a number of spelling correction programs, as described in chapter 3. Whilst in most cases this method has been used for single error misspellings, Backhouse describes how it would be used for correction of multiple error misspellings: for detail of his algorithm and Pascal implementation see chapter 5 (Backhouse, 1979). His method will be described here, in less detail.

The task is to find the best repair for transforming a string **E** (the error) into string **C** (the correction). The edit operations that are used to do the transformation are:

- \* insert a character
- \* delete a character
- \* change a character

\* transpose two adjacent characters<sup>1</sup>

Additionally, leaving a letter unchanged can be considered an operation with no effect.

The edit operations will be indicated as follows:

b -> 0	delete 'b' from the error string
0 -> d	insert 'd' in the error string
b -> d	change 'b' to 'd'
ei -> ie	transpose 'ei' to 'ie'
a -> a	leave 'a' unchanged

There are various ways in which any misspelling can be edited to form the correction. For example, some of the ways to transform 'ricev' to 'receive' are:

1. r -> r	2. r -> r	3. r -> r	4. r -> 0
i -> e	i -> 0	i -> e	i -> 0
c -> c	0 -> e	c -> c	c -> 0
e -> e	c -> c	e -> e	e -> 0
0 -> i	0 -> e	v -> i	v -> 0
v -> v	0 -> i	0 -> v	0 -> r
0 -> e	ev -> ve	0 -> e	0 -> e
			0 -> c
			0 -> e
			0 -> i
			0 -> v
			0 -> e

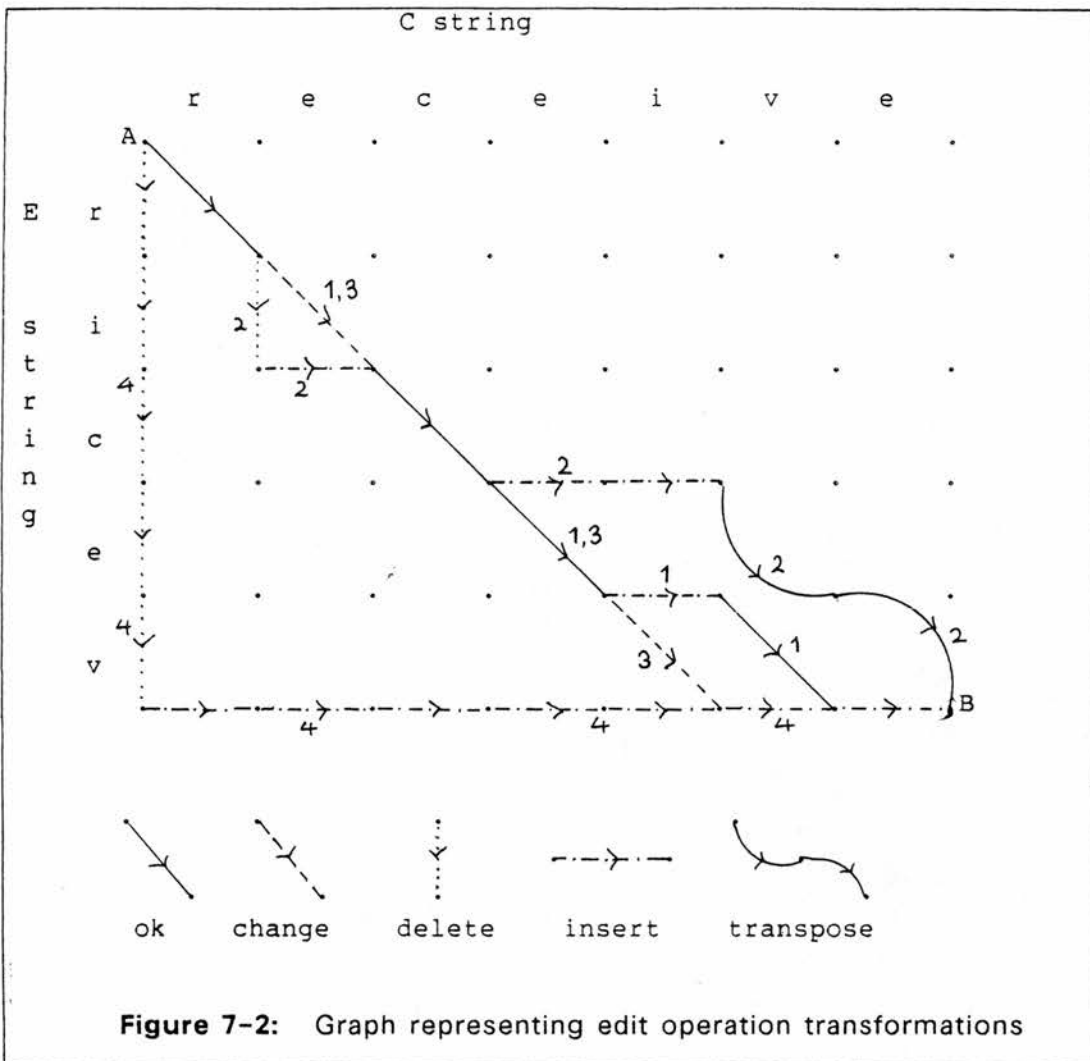
In the fourth example all the characters in the E string are deleted and all those in the C string are inserted.

These different transformations may be represented as a graph, where each node corresponds to a position in E string and a position in C string, and each arc corresponds to an edit operation.

The graph in figure 7-2 represents the transformations 1. to 4. above. It can be seen that there are many possible paths through the graph. In fact, any one word can be transformed into any other word by deletion of all the error string characters and insertion of all the correction string characters. If we assign costs to each edit operation then the cost of any path through the graph (i.e.

---

<sup>1</sup>Transposition can only be made between adjacent letters: repeated transpositions are not permitted.

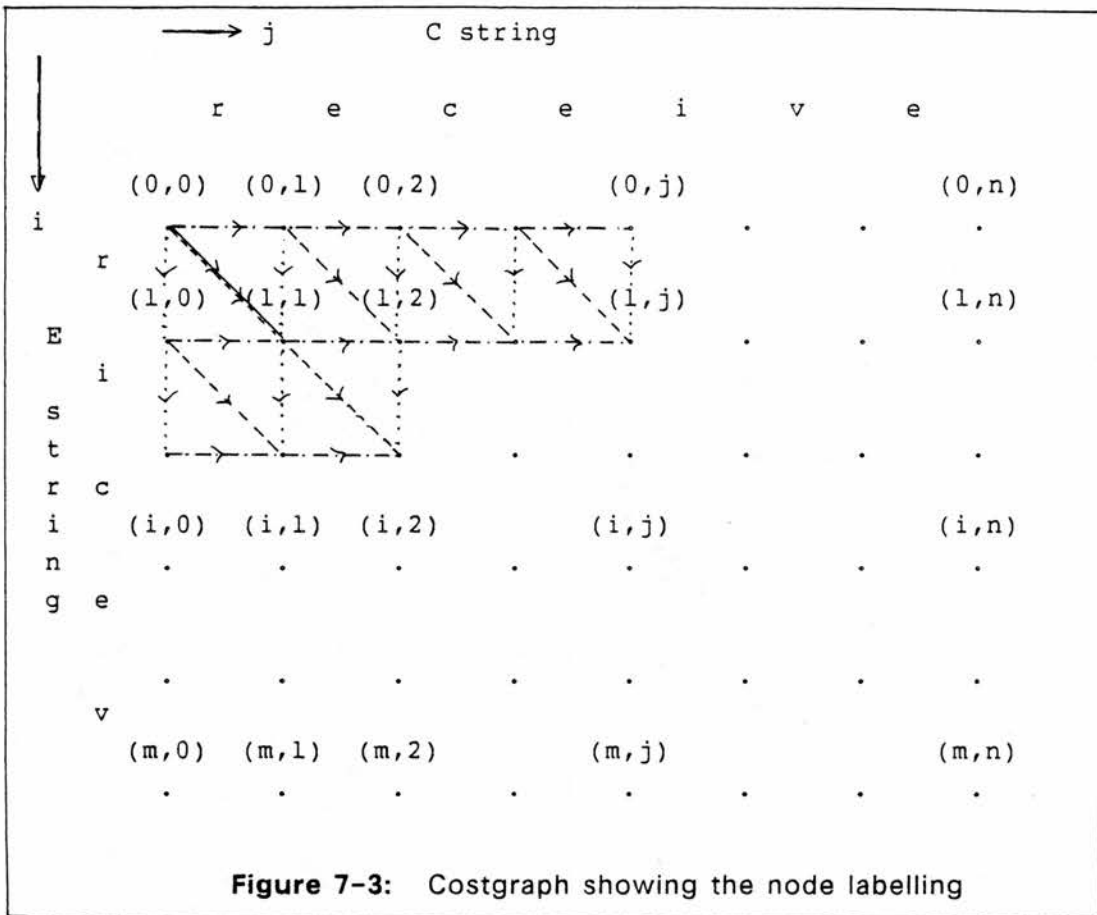


the cost of any particular transformation or sequence of edit operations) is the sum of all operations on that path. The 'best repair' is taken to be the path through the graph, from A to B, with least total cost. For example, if unit cost were assigned to all edit operations, and zero cost to ok(no change), then the least cost path in the above example would be path 1, with a cost of 3 units.

#### To calculate the least cost path

Consider two strings E and C, with lengths m and n respectively. The graph representing the transformations between these strings will have dimensions (0 to m) by (0 to n) where the first letters of each string E(1) and C(1) are associated with the node (1,1) (see figure 7-3).

The least cost path from the node (0,0) to any point in the graph can be calculated. For points (0,1), (0,2)...(0,n) the least cost will be the cost of inserting letters C(1), C(2), C(3)...C(n). The least cost path from (0,0) to (0,1) will be:



```
cost(ins(C(1)))
```

which is the cost of inserting the character that is in position 1 in the C string (inserted in the E string).

Similarly the least cost path to (0,2) will be:

$$\text{cost}(\text{ins}(C(1))) + \text{cost}(\text{ins}(C(2)))$$

This will be referred to as the minimum cost path to the node (0,2), or mincost(0,2). Thus:

```
mincost(0,1) = cost(ins(C(1)))
mincost(0,2) = cost(ins(C(1))) + cost(ins(C(2)))
              = mincost(0,1) + cost(ins(C(2)))
```

So for any node  $(0,j)$  in the path  $(0,0)$  to  $(0,n)$ :

$$\text{mincost}(0,j) = \text{mincost}(0,j-1) + \text{cost}(\text{ins}(C(j)))$$

For any node  $(i,0)$  in the path  $(0,0)$  to  $(m,0)$ :

$$\text{mincost}(i,0) = \text{mincost}(i-1,0) + \text{cost}(\text{del}(E(i)))$$

where  $\text{cost}(\text{del}(E(i)))$  is the cost of deleting the  $i$ th character of the  $E$  string, the error string.

Considering the node  $(1,1)$  there are four possible paths to this node:

1. delete( $E(1)$ ) and insert( $C(1)$ )

insert( $C(1)$ ) and delete( $E(1)$ )

change  $E(1)$  to  $C(1)$

if  $E(1)$  and  $C(1)$  are the same leave them unchanged

The respective costs for each of these will be:

$$(i) \text{mincost}(0,1) + \text{cost}(\text{del}(E(1)))$$

$$\text{mincost}(1,0) + \text{cost}(\text{ins}(C(1)))$$

$$\text{mincost}(0,0) + \text{cost}(\text{change}(E(1),C(1)))$$

$$\text{mincost}(0,0) + \text{cost}(\text{ok}(E(1),C(1)))$$

The  $\text{mincost}(1,1)$  will be the minimum of these four costs.

The least cost for any node  $(1,j)$  in the path  $(1,0)$  to  $(1,n)$  is, therefore:

$$\begin{aligned} \text{mincost}(1,j) = \text{minimum of} \\ &(\text{mincost}(0,j) + \text{cost}(\text{del}(E(1))), \\ &\text{mincost}(1,j-1) + \text{cost}(\text{ins}(C(j))), \\ &\text{mincost}(0,j-1) + \text{cost}(\text{change}(E(1),C(j))), \\ &\text{mincost}(0,j-1) + \text{cost}(\text{ok}(E(1),C(j)))). \end{aligned}$$

and similarly for any node  $(j,1)$  in the path  $(0,1)$  to  $(m,1)$

$$\begin{aligned} \text{mincost}(i,1) = \text{minimum of} \\ &(\text{mincost}(i-1,1) + \text{cost}(\text{del}(E(i))), \\ &\text{mincost}(i,0) + \text{cost}(\text{ins}(C(1))), \\ &\text{mincost}(i-1,0) + \text{cost}(\text{change}(E(i),C(1))), \\ &\text{mincost}(i-1,0) + \text{cost}(\text{ok}(E(i),C(1)))). \end{aligned}$$

For other nodes in the graph, where  $i \geq 2$  and  $j \geq 2$ , the cost of transposition must also be considered:

$\text{mincost}(i,j)$ , where  $i \geq 2$  and  $j \geq 2$ ,  
 = minimum of  
 $(\text{mincost}(i-1,j) + \text{cost}(\text{del}(E(i))))$ ,  
 $\text{mincost}(i,j-1) + \text{cost}(\text{ins}(C(j)))$ ,  
 $\text{mincost}(i-1,j-1) + \text{cost}(\text{change}(E(i),C(j)))$ ,  
 $\text{mincost}(i-1,j-1) + \text{cost}(\text{ok}(E(i),C(j)))$ ,  
 $\text{mincost}(i-2,j-2) + \text{cost}(\text{transp}(E(i-1),E(i)))$ .

The cost of the least cost path through the graph (i.e. the minimum cost repair of the two strings) is the minimum cost path from node (0,0) to node (m,n),  $\text{mincost}(m,n)$ :

$\text{mincost}(m,n)$  = minimum of  
 $(\text{mincost}(m-1,n) + \text{cost}(\text{del}(E(m))))$ ,  
 $\text{mincost}(m,n-1) + \text{cost}(\text{ins}(C(n)))$ ,  
 $\text{mincost}(m-1,n-1) + \text{cost}(\text{change}(E(m),C(n)))$ ,  
 $\text{mincost}(m-1,n-1) + \text{cost}(\text{ok}(E(m),C(n)))$ ,  
 $\text{mincost}(m-2,n-2) + \text{cost}(\text{transp}(E(m-1),E(m)))$ .

Backhouse discusses the use of the algorithm for spelling correction, assigning unit cost to each edit (Backhouse, 1979). Here, however, different costs have been assigned to each edit operation, and also varying costs are assigned according to the particular characters involved in the edit. Weightings are assigned according to the particular edit operation and character, and from these weightings the costs are calculated. (Details of the calculations and weightings are given below). Thus for any error the minimum cost of transforming it to match any word in the dictionary can be calculated. For any word input by the user, those words from the shortlist with lowest 'minimum cost repair' are selected, and offered as options for correction to the user. Three examples of the editcost program in use are given in figure 7-4.

#### 7.2.4. Relation of the children's errors

A major objective of the first study was to collect data on the errors made by children from the Reading Unit, when writing compositions. Details of this study are given in chapter 6. Depending upon the frequency of particular errors made by the children, weightings were assigned to edit operations in the editcost program. Those errors made most frequently were assigned highest weighting, and therefore lowest cost, which influenced the selection of the minimum cost repair of a misspelling.

w:check  
 What word do you want to check?  
 w:berayd  
 Wait a minute while I check it  
 I have nearly finished checking it  
 It could be

buried

bed

bad

board

w:check  
 What word do you want to check?  
 w:houes  
 Wait a minute while I check it  
 I have nearly finished checking it  
 It could be

house

hours

horses

hour

w:check  
 What word do you want to check?  
 w:wen  
 Wait a minute while I check it  
 It could be

when

went

win

we

**Figure 7-4:** Example of the spelling corrector in use

For each child, across sessions, the frequency of incorrect words was noted, together with the misspelling and correction. For each misspelling, all errors within the misspelt word were treated separately, and were classified according to whether they involved deletion, insertion, transposition, or the changing of a

character. The transformation of spelling to correction involving the minimum number of edit operations was chosen in most cases, using the subjective judgement of the author to decide the most likely error in cases where a number of interpretations might be made.

These judgements were based on previous observation of the children's errors, and reported 'frequent errors' of other researchers. For example, a preference in certain cases for phonetic over non-phonetic substitutions. Three examples of the choices made will be given here. In the case of 'panes' misspelt as 'pains' two possible transformations, each involving two edit operations, are:

- |           |           |
|-----------|-----------|
| 1. p -> p | 2. p -> p |
| a -> a    | a -> a    |
| i -> n    | i -> 0    |
| n -> e    | n -> n    |
| s -> s    | 0 -> e    |
|           | s -> s    |

The second is chosen as the interpretation: the confusion of 'ai' for 'a\_e' was considered a more likely error than the confusion of 'i' and 'n' and of 'n' and 'e'.

Similarly, in 'wigule' for 'wiggled' the second example is chosen:

- |           |           |
|-----------|-----------|
| 1. w -> w | 2. w -> w |
| i -> i    | i -> i    |
| g -> g    | g -> g    |
| u -> g    | 0 -> g    |
| l -> l    | u -> 0    |
| e -> e    | l -> l    |
| 0 -> d    | e -> e    |
|           | 0 -> d    |

In this case it seemed more likely that the second 'g' had been omitted and that the 'u' was part of the 'ule' grapheme, rather than that 'u' and 'g' had been confused. In some cases, where the transformation was not clear, both sets of errors were counted:

'lifet' for 'left'

- |           |           |
|-----------|-----------|
| 1. l -> l | 2. l -> l |
| i -> e    | i -> 0    |
| f -> f    | fe -> ef  |
| e -> 0    | t -> t    |
| t -> t    |           |

A frequency count was made of each operation involved in an error, and the particular letter(s) involved. Totals for errors, according to the edit operation involved, are given in figure 7-5.

1. INSERT.	159	(43%)
2. DELETE.	78	(21%)
3. TRANSPOSE.	24	(7%)
4. CHANGE.	109	(29%)
5. TOTAL.	370	

**Operations:**

insert - insertion of a letter into error to get correction.

delete - deletion of a letter from error to give correction.

change - change of letter in error into corresponding letter in correction, error/correction.

transpose - transposition of letters AB in error to BA in correction.

**Figure 7-5: Study 1: frequency of error types**

It can be seen that the most frequent errors were those requiring the insert operation for correction i.e. letters were omitted from the correct word. The change operation accounted for the next highest proportion of errors, with deletions having slightly lower percentage occurrence. A much smaller percentage of errors made involved transpositions. These findings are in accord with those of Masters (Masters, 1927) who also found insertions most frequent and transpositions the least frequent.

Frequencies of errors for specific characters, for all children, are given in figure 7-6. The number of children making each error (maximum 7) is also given. It should be noted that, as well as considering the application of edit operations to individual characters (a to z), a number of special cases were also considered e.g. each 'l' of 'll' (double 'l' in error or in correction); silent initial 'k' or 'w' ('know' or 'write'); final 'e' ('cane'); silent second 'h' ('when'); the 'c' or 'k' of 'ck' ('back').

The most frequent errors were those involving vowels, 'e' in particular is involved in some 100 of the 370 errors.

Insert		Delete		Change		Transpose	
letter	freq.	letter	freq.	letters	freq.	letters	freq.
e	20/7	a	15/6	e/a	8/5	ae/ea	5/4
fnl e	20/6	fnl e	14/6	o/a	8/4	hg/gh	2/2
i	14/6	i	10/4	u/o	6/4	se/es	3/1
a	12/5	e	7/5	e/u	4/4	de/ed	2/1
r	12/5	o	5/3	a/e	4/4	ed/de	1/1
w	5/5	u	4/3	a/o	4/3	el/le	1/1
dbl l	7/4	r	4/3	s/c	3/3	le/el	1/1
scd h	7/4	scd h	4/2	m/n	3/3	er/re	1/1
g	7/3	g	2/2	d/b	3/2	en/ne	1/1
u	5/3	t	2/1	c/k	2/2	ol/lo	1/1
l	4/3	y	1/1	i/e	2/2	ye/ey	1/1
d	4/3	dbl r	1/1	a/u	2/2	uo/ou	1/1
o	4/3	dbl p	1/1	i/y	2/2	nh/hn	1/1
y	3/3	dbl g	1/1	t/p	2/2	fe/ef	1/1
t	5/2	dbl o	1/1	u/e	2/2	oh/ho	1/1
n	5/2	h	1/1	h/i	2/2	th/ht	1/1
c	3/2	kn k	1/1	k/c	3/1		
dbl s	2/2	n	1/1	f/v	3/1		
f	2/2	l	1/1	o/u	3/1		
dbl t	2/2	dbl n	1/1	c/g	2/1		
ck c	2/2	d	1/1	g/c	2/1		
s	2/2			s/k	2/1		
dbl r	2/2			o/e	1/1		
dbl e	1/1			u/w	1/1		
dbl d	1/1			t/r	1/1		

Frequency of errors: frequencies given as the number of errors made/number of the students making the error.

(Only 25 most frequent given for insert and change)

#### Abbreviations:

fnl = final (fnl e in make)      dbl = double (dbl l in bell)

scd = second (scd h in when)      kn = silent k (knit)

ck = c or k of pair ck

**Figure 7-6:** Study 1: frequency of types of spelling errors

### 7.2.5. Detail of the editcost program

The word to be checked, the *inpw*, is compared with each word on the shortlist, *dictw*, in turn. For each pair, *inpw* and *dictw*, the string to string repair graph is constructed and all cost paths through the graph are calculated, using the algorithm described above. Costs are determined by weightings assigned by the program at the start of each session (see below for details of the

weightings and calculations). The five words with minimum repair costs are always saved. When all the dictws on the shortlist have been compared with the inpw, the four with minimum repair cost are offered to the user. Examples of the costs of the five saved dictws for each of four inpws are given in figure 7-7, together with the intended word.

inpw=reck	wreck
Options were	
wreck cost= 0.362	
rock cost= 0.883	
reach cost= 1.211	
rocks cost= 1.433	
recall cost= 1.573	
inpw=roack	rock
Options were	
rock cost= 0.466	
rocks cost= 1.016	
wreck cost= 1.319	
reach cost= 1.733	
road cost= 1.766	
inpw=kuver	cover
Options were	
cover cost= 1.082	
curve cost= 1.677	
keeper cost= 1.733	
corner cost= 2.293	
keep cost= 2.316	
inpw=bilt	built
Options were	
built cost= 0.407	
belt cost= 0.550	
bit cost= 0.764	
belts cost= 1.100	
build cost= 1.290	

**Figure 7-7:** Example of candidates and minimum editcost

Costs were assigned according to the particular character and edit operation involved. As well as considering the characters 'a' to 'z', in a number of cases the position of the character, and adjacent characters, were also considered. These cases are referred to as 'special cases', and were assigned weightings

independent of those assigned to the same characters in other contexts. For example, 'e' in the final position in a string (error or correction) had a different weighting from 'e' in other positions. A list of special cases is given in figure 7-8. When considering any character in the E or C strings, it was first tested to see if it was one of these 'special cases'. If it was, it was recoded with an upper case character, as shown in the figure 7-8. For example, the 'k' in 'know' was recoded as 'B'.

Representation	Letter(adjacent letter)
A	final e
B	k(n) at start of word
C	w(r) or w(h)
D	h in second position
E	t(t)
F	r(r)
G	o(o)
H	e(e)
I	p(p)
J	b(b)
K	k(k)
L	m(m)
M	l(l)
N	s(s)
O	f(f)
P	n(n)
Q	d(d)
R	z(z)
S	c(k)
T	(c)k
U	g(h)

**Figure 7-8:** Representation of 'special cases'

Weightings assigned were represented as array values in the range 2.5 to 10. Four arrays of weightings were set up, one for each edit operation (no weighting was recorded for the ok edit). A list of the actual weightings assigned is given in figure 7-9. The weighting for deleting or inserting any character is stored in the one-dimensional arrays deletewtarray and insertwtarray, respectively, indexed by the character itself (or character representation in special cases). The changewtarray and transposewtarray are two-dimensional. Weightings are given in the figure for a number of change and transpose combinations. The cost of changing pairs of characters is considered

separately in a number of cases: e.g. the cost of changing 'f' to 'th', or 'ff' to 'gh' are indexed separately. (see figure 7-9).

For each pair of nodes to be compared the costgraph is constructed. At each node the pair of characters  $E(i)$  and  $C(j)$  is considered: they are tested first to see if they are 'special cases'. Weightings for each edit operation at the node  $(i,j)$  are found. Insert weightings and delete weightings are the values of the array elements indexed by the characters. If there is a match for a transposition, that is if  $E(i-1)=C(j)$  and  $E(i)=C(j-1)$ , then the transpose edit weighting is given by the tranposewtarray element  $(E(i-1),E(i))$ . If  $E(i)=C(j)$ , that is if the characters match, then the edit cost is zero, otherwise the change edit weighting is found. Characters preceding and following  $E(i)$  and  $C(j)$  are noted. Adjacent pairs of characters,  $E(i),E(i+1)/E(i-1),E(i)/C(j),C(j+1)/C(j-1),C(j)$ , are compared with the characters pairs listed under 'changewts' (see figure 7-9) and if a match is found weightings are assigned accordingly: otherwise the value stored in the changewtarray element  $(E(i),C(j))$  is the change edit weighting for the current node. The weightings are used to calculate the cost for each edit operation at each node, where

$$\text{cost} = 0.05 + (2.5/\text{weighting})$$

The maximum cost of a single edit is 1.05 (weighting=2.5) and the minimum cost is 0.3 (weighting=10). Using the algorithm described above, the minimum cost of each node is calculated, for all nodes, and the least cost path through the graph determined.

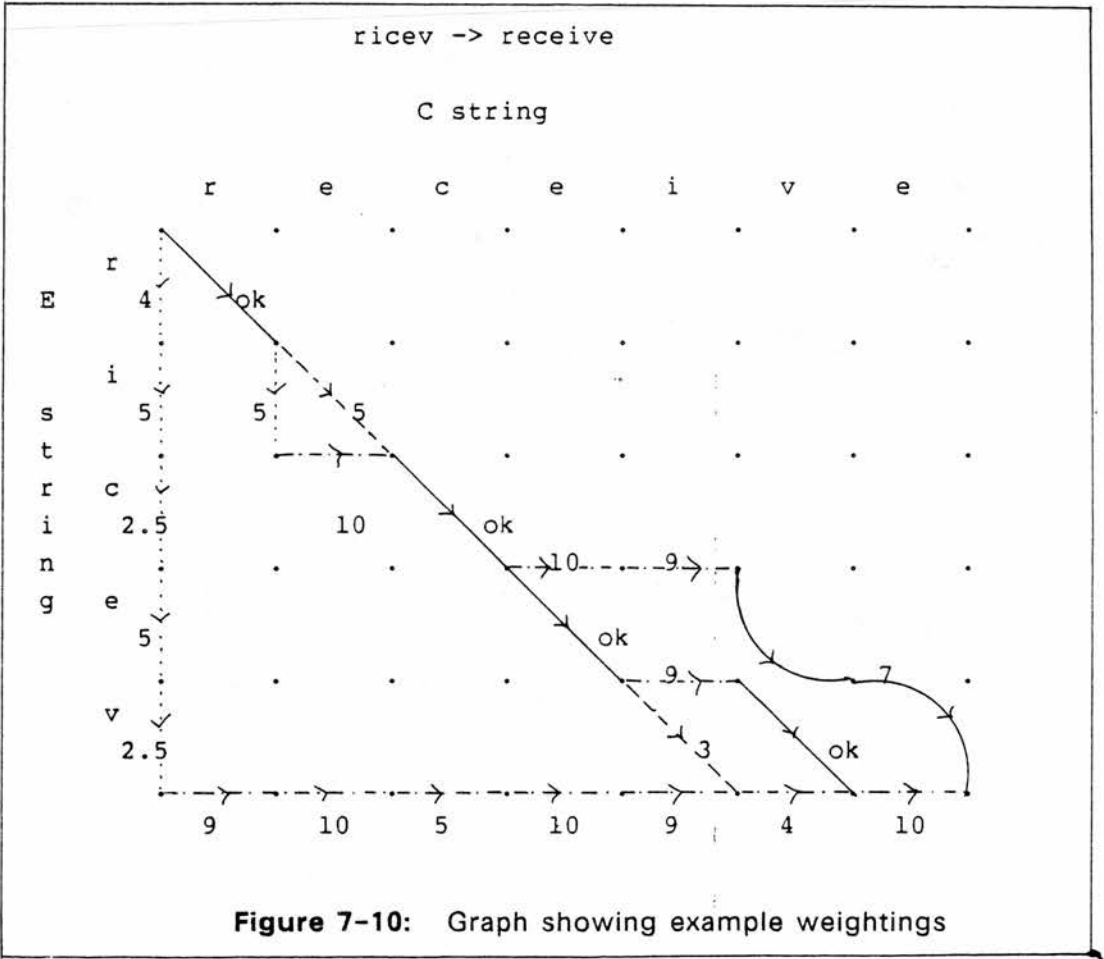
The weightings were chosen by considering those errors most frequently made by the children in Study 1, and by experimenting with different costs and weighting functions, testing them on a subset of the errors. They were set up such that 2 very frequent errors, that might often occur together, would have lower cost than a single (less likely error), and hence the matched dictw involving 2 edit operations could be chosen in preference to another dictw involving only 1 edit operation. For example, if the misspelling is "wud" it could be edited to "would" by inserting "o" and "l" (2 operations, cost=.814). It could be edited to match "mud" with 1 operation, change "w" to "m" (cost=.838). The match to "would" is cheaper. In theory, 3 of the 'cheapest' edits (minimum cost 0.3, weighting 10) would be less costly than 1 of the most expensive (maximum

insertwt	deletewt	changewt	transwt
a 9.0	a 6.0	a/e 7.0	ae 9.0
b 4.0	b 2.5	a/o 7.0	au 8.0
c 5.0	c 2.5	a/u 5.0	de 8.0
d 7.0	d 3.5	b/d 4.0	ea 9.0
e 10.0	e 5.0	c/k 5.0	ei 9.0
f 5.0	f 2.5	e/a 8.0	el 9.0
g 7.0	g 4.0	e/i 4.0	es 9.0
h 8.0	h 3.5	e/u 7.0	er 9.0
i 9.0	i 5.0	h/i 5.0	ed 8.0
j 4.0	j 2.5	i/e 5.0	ey 8.0
k 4.0	k 2.5	i/y 6.0	ef 8.0
l 7.0	l 3.5	k/c 4.0	fe 8.0
m 4.0	m 2.5	m/n 6.0	gh 8.0
n 5.0	n 3.5	n/m 4.0	hg 8.0
o 7.0	o 5.0	o/u 4.0	hn 8.0
p 4.0	p 2.5	p/b 6.0	ht 8.0
q 4.0	q 2.5	s/c 6.0	ie 9.0
r 9.0	r 4.0	t/p 5.0	le 9.0
s 5.0	s 3.5	u/a 4.0	nh 8.0
t 5.0	t 3.5	u/e 5.0	re 9.0
u 7.0	u 5.0	u/o 7.0	se 9.0
v 4.0	v 2.5	y/i 4.0	th 8.0
w 8.0	w 2.5	f/gh 4.0	ua 8.0
x 4.0	x 2.5	f/th 6.0	ye 8.0
y 7.0	y 4.0	f/ph 6.0	all others=7.0
z 4.0	z 2.5	g/ch 4.0	
A 10.0	A 6.0	j/ch 4.0	
B 8.0	B 3.5	v/th 4.0	
C 8.0	C 2.5	w/gh 4.0	
D 8.0	D 4.0	y/gh 4.0	
E 7.0	E 3.5	y/ie 5.0	
F 9.0	F 4.0	ch/j 4.0	
G 7.0	G 5.0	ch/t 4.0	
H 10.0	H 5.0	ff/gh 4.0	
I 6.0	I 3.5	gh/ff 3.5	
J 6.0	J 3.0	oo/ue 4.5	
K 7.0	K 4.0	ow/ue 4.5	
L 6.0	L 3.0	th/f 4.5	
M 8.0	M 3.5	all others=3.0	
N 7.0	N 3.5		
O 6.0	O 3.0		
P 7.0	P 4.0		
Q 7.0	Q 3.5		
R 6.0	R 3.0		
S 7.0	S 3.0		
T 6.0	T 3.0		
U 4.0	U 4.0		

Figure 7-9: Weightings assigned to edit functions

cost 1.05, weighting=2.5): it is very difficult, however, to imagine cases when this would happen. Though the transposition errors were the least frequent of the error types they were given high weightings: this was to increase the likelihood of any error being classed as a transposition, if it could be considered as such.

An example will be given to illustrate the method of calculation of the minimum cost repair. For the four paths given in figure 7-2, the graph showing the weightings, four of the possible transformations, and the costs for each transformation are given. The weightings for the transformations are given in figure 7-10.



The transformations are:

1. r -> r	2. r -> r	3. r -> r	4. r -> 0
i -> e	i -> 0	i -> e	i -> 0
c -> c	0 -> e	c -> c	c -> 0
e -> e	c -> c	e -> e	e -> 0
0 -> i	0 -> e	v -> i	v -> 0
v -> v	0 -> i	0 -> v	0 -> r
0 -> e	ev -> ve	0 -> e	0 -> e
			0 -> c
			0 -> e
			0 -> i
			0 -> v
			0 -> e

The costs for each transformation will be:

$$1. 0 + .55 + 0 + 0 + .327 + 0 + .3 = 1.177$$

$$2. 0 + .55 + .3 + 0 + .3 + .327 + .407 = 1.884$$

$$3. 0 + .55 + 0 + 0 + .883 + .675 + .3 = 2.408$$

$$4. .675 + .55 + 1.05 + .55 + 1.05 + .327 + .3 + .55 + .3 + .327 + .675 + .3 = 6.654$$

The performance of the editcost algorithm, together with possible improvements and extensions, is discussed in chapter 8.

### 7.3. Phonemic coding of words: the phoncode program

#### 7.3.1. General overview

When phoneme-grapheme correspondence rules are used to generate spellings, misspellings may occur when incorrect rules are used, or when the rules are inappropriate (for example an irregularly spelt word). If a word is misspelt in this way, it might be corrected as follows:

1. isolate the graphemes in the word
2. infer which phoneme-grapheme correspondence rules may have been used to produce the graphemes
3. select a set of phonemes that could have generated these graphemes
4. use the set of phonemes to generate the correct spelling

In the first three steps, the phoneme-grapheme correspondence rule is being used in reverse. This is not equivalent, however, to using the grapheme-phoneme rules as in reading. This method of correction is the basis of the phoncode program, to be described here.

### 7.3.2. Related work

Research on grapheme to phoneme correspondence rules is of interest here, in that it gives some guide to pronunciation of correctly spelt words, and also for regular non-words. As discussed in chapter 2, grapheme-phoneme rules can be specified more easily than phoneme-grapheme rules. Work by Venezky is of particular relevance here (Venezky, 1966). The grapheme-phoneme rules are of limited use, however, since they do not provide information about all possible phonemes that might be generated; or which alternative graphemes could be used in place of the given one, in order to produce the same phoneme.

Work on computer text-to-speech production systems is also of relevance (Ellovitz et al, 1976), (Ciarcia, 1982), (Allen, 1981), but has similar limitations. The focus is on specifying grapheme-phoneme rules in certain graphemic contexts, and most difficulties are encountered in these systems with pronunciations of irregularly spelt words.

Research on phoneme-grapheme correspondence is of more relevance here, in particular the studies by Hanna et al (Hanna et al, 1966), and by Simon and Simon (Simon and Simon, 1973). These studies are discussed in Chapter 2. Information about the correct graphemes for representing phonemes in specific words was obtained in these studies. The phoneme-grapheme correspondences that were generated, used in reverse, would provide some of the information required for the phoncode program. However, as with Venezky's work, and the text-to-speech research, legitimate correspondences (phoneme-grapheme and grapheme-phoneme) in specific contexts were being studied: the aim was to produce correct pronunciation or spelling, using context-specific rules. As the concern in this thesis is misspellings, the correspondence rules to be used must be largely context-free. A misspelling might well be a 'correct' phoneme-grapheme correspondence used in an 'incorrect' context. Therefore, information from these studies is useful, but not sufficient.

### 7.3.3. Design of the phoncode program

The phoncode program takes an input word (inpw) and selects those words from the dictionary that can be considered phonemically equivalent to it. The dictionary words matched are offered to the user as possible corrections for the inpw. See figure 7-11 for some examples.

```

: wear
  Is it one of these?
    wear
    where
    were

: turtul
  Is it one of these?
    turtle

: thiar
  Is it one of these?
    their
    there

: somb
  Is it one of these?
    some
    sum
  
```

**Figure 7-11:** Examples of use of the phoncode program

The phoncode program works by segmenting the inpw into graphemes; finding phonemes that these graphemes might represent (in any context); using a set of these phonemes to search for legitimate words represented by matching phonemes. Decisions were made concerning selection of the set of graphemes (into which words were segmented), the set of phonemes (used to represent all words), and the grapheme-phoneme correspondences. Information from four sources was used in making these decisions:

- \* errors made by children in both studies;
- \* related work by Hanna et al (Hanna et al, 1966);
- \* related work by Simon and Simon (Simon and Simon, 1973);
- \* additional examples generated by the author.

### 7.3.4. Defining the set of phonemes

In the Hanna study words were phonemically coded using a 62-phoneme classification scheme, based on the Merriam-Webster dictionary pronunciation code (Hanna et al, 1966). This scheme used 32 vowel phonemes and 30 consonant phonemes. Later they reduced the number of vowel phonemes in the classification scheme, combining the weakest vowel forms to the schwa<sup>2</sup>, and combining several other categories where distinctions between phonemes were not clear, e.g. reducing the two categories /A/ and /A1/ (ale and chaotic) to a single category, /A/. The total number of vowel phoneme categories was reduced to 22. Ellovitz et al. (Ellovitz et al, 1976) adopted a scheme of 41 phonemes, 16 vowel phonemes and 25 consonant phonemes, in their text-to-speech program. The phonemes were used in converting english text to IPA representation<sup>3</sup>, which is then passed to the Votrax synthesizer (Ellovitz et al, 1976). Morris-Wilson, in his phonemic transcription textbook, used a representation of 44 phonemes, 24 consonantal and 20 vowel phonemes, taken from 'Gimson's Set' (see (Morris-Wilson, 1984) for details and references).

The set used in this study comprised 46 phonemes in all, 20 vowel phonemes and 26 consonantal phonemes. Figures 7-12 and 7-13 show the phonemes used, with examples<sup>4</sup>.

The 20 vowel phonemes were equivalent to 20 of the vowel phonemes used by Hanna, combining the /E5/ and /U2/ categories to match completely. Of the vowel phonemes, 19 of the 20 used by Morris-Wilson were used here. The /U2/ category was omitted (combined with the /o:/ category), and a separate vowel phoneme /ju/ was added: Morris-Wilson has a consonant category /j/, but no independent vowel category. Two consonantal categories were added to the Morris-Wilson set to make the set here: /ks/ and /kw/ are added, as they were considered to be single phonemes in certain contexts. This set is equivalent to

---

<sup>2</sup>the vowel sound occurring in unstressed syllables e.g. about, vowel, mother

<sup>3</sup>International Phonetic Alphabet

<sup>4</sup>The code column indicates the integer code used in the actual program: the text notation will generally be used in this thesis for ease of reading.

Examples	Morris- Wilson	IPA	Hanna et al	Notation here (text) (code)	
late, day	eI	eI	A/A1	<b>el</b>	10
air, care	eə	ɛə	A2	<b>eE</b>	12
bat, add	æ	æ	A3/A4/A6	<b>ae</b>	13
car, aunt	ɑ:	ɑ	A5	<b>a:</b>	15
about, silent	ə	ə	A7/E4/I4 O4/U4	<b>E</b>	57
beat, keep	i:	i	E/E1	<b>i</b>	50
here, ear	Iə	Iə	E2	<b>IE</b>	52
end, let	e	ɛ	E3	<b>e</b>	53
maker, urn	ɜ:	ɑ	E5/U2	<b>e:</b>	55
ice, high	aI	aI	I	<b>al</b>	90
ill, bit	I	I	I3	<b>l</b>	93
boat, know	əʊ	oʊ	O/O1	<b>EU</b>	150
port, saw	o:	o	O2	<b>o:</b>	152
pot, soft	ɒ	ɒ	O3/O5	<b>O</b>	153
food, rude	u:	u	O6	<b>u</b>	156
foot, book	ʊ	ʊ	O7	<b>U</b>	157
cube, unite	-	ju	U/U1	<b>ju</b>	200
up, son	^	^	U3	<b>^</b>	203
oil, boy	I	I	OI	<b>ol</b>	158
out, cow	aU	aʊ	OU	<b>aU</b>	159
honest	-	-	H9	-	-
late	-	-	E9	-	-

**Figure 7-12:** Representation of vowel phonemes

the Hanna set with category pairs /L/ and /L1/, /M/ and /M1/, /N/ and /N1/, /W/ and /WH/ each being considered as one category. The basis for combination of phoneme categories, considered distinct by Hanna et al, was that in many contexts the pronunciations would be considered indistinguishable to a non-linguist, both when pronounced by the author, or by the children from the reading unit.

### 7.3.5. Phoneme-grapheme correspondences

When considering a word and its phonemic representation, how do we decide which character or characters represent each phoneme?<sup>5</sup> In cases such as the following the relationship is clear:

<sup>5</sup>Note that the convention here will be grapheme = /phoneme/.

Examples	Morris-	IPA	Hanna	Notation here	
	Wilson		et al	(text)	(code)
bad, rub	b	b	B	<b>b</b>	20
bad, day	d	d	D	<b>d</b>	40
fat, rough	f	f	F	<b>f</b>	60
go, big	g	g	G	<b>g</b>	70
hit, behind	h	h	H	<b>h</b>	80
gin, joke	dʒ	dʒ	J	<b>dz</b>	100
keep, cock	k	k	K	<b>k</b>	110
loud, kill	l	ɫ	L, L1	<b>l</b>	120
mad, jam	m	m	M, M1	<b>m</b>	130
man, no	n	n	N, N1	<b>n</b>	140
pit, top	p	p	P	<b>p</b>	160
run, bread	r	ɹ	R	<b>r</b>	170
sit, loss	s	s	S	<b>s</b>	180
trap, step	t	t	T	<b>t</b>	190
very, love	v	v	V	<b>v</b>	210
wash, when	w	w	W, HW	<b>w</b>	220
yellow, yet	j	j	Y	<b>y</b>	230
zoo, beds	z	z	Z	<b>z</b>	240
chair, lunch	tʃ	tʃ	CH	<b>ch</b>	31
ethics, accent	-	-	KS	<b>ks</b>	111
quick, aqua	-	kw	KW	<b>kw</b>	112
sing, along	ŋ	ŋ	NG	<b>ng</b>	142
sugar, bush	ʃ	ʃ	SH	<b>sh</b>	181
theatre, thank	θ	θ	T1	<b>th</b>	191
that, with	ð	ð	T2	<b>tv</b>	192
garage, pleasure	ʒ	ʒ	ZH	<b>zh</b>	241
honest	-	-	H9	-	-

**Figure 7-13:** Representation of consonant phonemes

cat = /k/ /ae/ /t/  
 dog = /d/ /O/ /g/  
 biting = /b/ /aI/ /t/ /I/ /ng/

c = /k/	d = /d/	b = /b/
a = /ae/	o = /O/	i = /aI/
t = /t/	g = /g/	t = /t/
		i = /I/
		ng = /ng/

In some other words it may be a little more difficult to decide, but an interpretation can be given:

abbey = /ae/ /b/ /i/  
 nation = /n/ /eI/ /sh/ /E/ /n/  
 science = /s/ /aI/ /E/ /n/ /s/

a = /ae/	n = /n/	sc = /s/
bb = /b/	a = /eI/	i = /aI/
ey = /i/	ti = /sh/	e = /E/
	o = /E/	n = /n/
	n = /n/	ce = /s/

With the following examples, however, it is much more difficult to decide which graphemes correspond to each phoneme:

lamb = /l/ /ae/ /m/

l = /l/	l = /l/
a = /ae/	a = /ae/
m = /m/	mb = /m/
b = ?	

receive = /r/ /E/ /s/ /i/ /v/

r = /r/	r = /r/	r = /r/
e = /E/	e = /E/	e = /E/
c = /s/	ce = /s/	ce = /s/
e = /i/	i = /i/	i+e = /i/
i = ?	ve = /v/	v = /v/
v = /v/		
e = ?		

vague = /v/ /eI/ /g/

v = /v/	v = /v/	v = /v/
a = /eI/	a = /eI/	a = /eI/
g = /g/	gu = /g/	gue = /g/
ue = ?	e = ?	

It is necessary to specify the full set of graphemes such that each may be considered to correspond to a single phoneme (or a number of alternative single phonemes). Hanna et al. (Hanna et al, 1966) and Simon and Simon (Simon and Simon, 1973) each give a set of phoneme-grapheme correspondences, the latter's set being a slight modification of the former's. Hanna et al list 107 different graphemes, and Simon and Simon list 104. Yannakoudakis and Fawthrop (Yannakoudakis and Fawthrop, 1983b) divide words into vowel and consonant elements, using 267 elements.

A number of specific problems were encountered when attempting to segment words into graphemes, some of the problems being illustrated in the examples above.

For example, consider 'silent letters', that is, single characters which apparently do not represent a phoneme. The final 'e' is a special case of this in words where it is used to modify the pronunciation of the preceding vowel (often referred to as the 'magic e' in teaching). It modifies the vowel, changing a soft vowel into a hard one:

can	=	/k/ /ae/ /n/	cane	=	/k/ /eI/ /n/
fin	=	/f/ /I/ /n/	fine	=	/f/ /aI/ /n/
cut	=	/k/ / <sup>^</sup> / /t/	cute	=	/k/ /ju/ /t/

The 'e' may be treated as a separate silent grapheme, or it may be considered part of the preceding vowel grapheme, represented in 'cane' as 'a\_e'. Hanna et al (Hanna et al, 1966) dealt with this problem by inventing a separate /E9/ phoneme category to represent the grapheme 'e'. If it is represented in this way, it is no longer linked to the preceding vowel. However, in some cases, where the 'e' is not final but still modifies the preceding vowel, the 'e' is sounded:

cases = /k/ /eI/ /s/ /I/ /z/

The 'e' corresponds to /I/. The graphemic construction in which the 'e' occurs will be described as a 'vc+e' grapheme (vowel, consonant + 'e'). In some cases the vowel and 'e' will be taken to correspond to a single phoneme, 'a\_e' in 'cane'; in others they represent two phonemes and are treated as separate graphemes.

Other silent letters cause difficulties, for example the 'b' in 'lamb', the 'k' in 'know', 'g' and 'h' in 'high', 'u' in 'guard'. Similarly in cases of double letters only one is pronounced: 'cotton', 'tell', 'pass'. The silent letter may be given no corresponding phoneme, or may be assigned to an adjacent grapheme with a corresponding phoneme. In the former case a letter that might be silent in some contexts could be placed anywhere in a word and not affect its pronunciation: if a phoneme /b0/ represented a silent 'b' then 'at' would be a possible misspelling of 'bat', or 'brubn' a misspelling of 'run'. If the silent letter is treated as part of another grapheme, then the cases in which it would be considered a legitimate phonetic misspelling are less arbitrary. If 'k' is silent

when followed by 'n', in the initial position of a word, the grapheme 'kn' could be considered as a possible correspondence to /n/, producing the following 'phonetic equivalents':

knot =	/n/ /O/ /t/	not =	/n/ /O/ /t/
kn =	/n/	n =	/n/
o =	/O/	o =	/O/
t =	/t/	t =	/t/

In discussing the silent letter graphemes, Hanna et al (Hanna et al, 1966) state that:

"Considerable disagreement occurs among linguists regarding how such graphemes should be classified." p. 14

They were all considered in this study to be part of a grapheme that had a corresponding phoneme: examples of graphemes incorporating these 'silent graphemes' are

'ce'	since	'gh'	ghost	'wh'	what
'wr'	write	'kn'	know	'bt'	debt

A complete list of the grapheme-phoneme correspondences used in this thesis is given in appendix B.

The schwa /E/ also presents problems, particularly in cases where it is followed by /n/:

opening = /EU/ /p/ /E/ /n/ /I/ /ng/

The 'en' is represented by two phonemes, /E/ and /n/. However, a possible misspelling of 'opening', where the 'e' is omitted, is 'opning'. To cater for this and similar cases, 'en' was treated as a grapheme representing a single phoneme, as was 'on' and 'an'.

In some dialects the 'r' in 'er' is not pronounced: the 'er' corresponds to a single phoneme, /e:/ or /E/. In other dialects the 'r' is sounded and the 'er' corresponds to two phonemes, /E/ and /r/. To allow for both cases 'er' was taken as a grapheme representing a single phoneme, or could be split to represent two phonemes. Other graphemes involving 'r' were treated in the same way:

ayor	oar	air	aer	are	ar
ear	re	er	ier	ir	our

oor ore or ure ur

Each corresponds to a single vowel phoneme, or to a single vowel and /r/. Some also correspond to the single phoneme /r/, or may be split further into more than two phonemes, e.g. 'ier' in 'carrier'. A large number of these correspondences were derived from the children's errors.

The phoneme /sh/ is spelt in a variety of different ways:

'ssi'	mission	'ss'	reassure	'ti'	nation
'si'	mansion	'ch'	machine	'sh'	show
'ci'	special				

All these were represented as individual graphemes.

A large number of graphemes correspond to a single phoneme in some cases, but to more than one phoneme in others. Because of this they must be taken to correspond to single phonemes but also able to be split into smaller graphemes, corresponding to other phonemes. For example, the grapheme 'ough':

cough = /k/ /O/ /f/  
 though = /tv/ /EU/

In 'cough' it corresponds to two phonemes; in 'though' only a single one. Doubled characters, however, and a number of other graphemes occur only as single phonemes in the majority of cases:

'igh'	high =	/h/ /aI/
'll'	will =	/w/ /I/ /l/
'ch'	chip =	/ch/ /I/ /p/
'ph'	phone =	/f/ /EU/ /n/

These graphemes, corresponding to one single phoneme, will be referred to as 'tied' graphemes. If a tied grapheme occurs in a word or a misspelling the corresponding phonemes were found, and the grapheme was segmented no further. Graphemes that are not tied were further segmented (or split) to find constituent graphemes and corresponding phonemes. Tied graphemes are indicated by a "1" preceding them in the grapheme-phoneme correspondence table (figure B-1, appendix B), and segmentable graphemes by a 0. All 'vc+e' graphemes are segmentable, and are indicated by \*.

### 7.3.6. Phonemic coding of the dictionary

The inpw was segmented into all possible graphemes, and phoneme correspondences were found for each grapheme. All words in the dictionary were coded phonemically. The set of phonemes representing the inpw were compared with those representing the dictionary words: if matches were found they were taken to be "possible phonemic equivalents".

Each word in the dictionary was hand coded by the author in conjunction with a linguist specializing in speech synthesis. The general guidelines suggested in the text "English Phonemic Transcription" (Morris-Wilson, 1984) were used in coding. 'Strong forms' were used in most cases (see p 76-82, (Morris-Wilson, 1984)) though the 'weak form' alternatives were also used in a few cases.

For a number of the words, the ambiguity of possible alternative pronunciations meant that more than one phonemic coding of a word was included in the coded dictionary. In particular, 'er' at the end of a word was coded both as /E/ and as /E/ /r/ to allow for variations in dialect. 'er' occurring elsewhere in a word was coded as /e:/ when representing 'r-less' dialects. Other 'vowel + r' graphemes were also coded with and without the 'r'.

```
board = /b/ /o:/ /r/ /d/
      & = /b/ /o:/ /d/
car = /k/ /a:/
     & = /k/ /a:/ /r/
error = /e/ /r/ /E/ /r/
      & = /e/ /r/ /E/
```

The 'd' in 'procedure', 'produce' and 'soldier' may be pronounced /d/ or /dz/, so both are represented:

```
produce = /p/ /r/ /O/ /d/ /ju/ /s/
      & = /p/ /r/ /O/ /dz/ /u/ /s/
```

Unstressed vowels were coded as schwa, /E/. In some words both the strong and the weak forms of vowels were coded:

```
around = /eI/ /r/ /aU/ /n/ /d/
around = /E/ /r/ /aU/ /n/ /d/
```

In general when there was uncertainty about the vowel phoneme, the /E/ was used and matched to all single vowel graphemes.

In some words, the schwa was omitted, and the following consonant only was coded. The schwa is unpronounced in some dialects.

```

difference = /d/ /I/ /f/ /E/ /r/ /E/ /n/ /s/
            & = /d/ /I/ /f/ /r/ /E/ /n/ /s/
edinburgh = /e/ /d/ /I/ /n/ /b/ /r/ /E/
            & = /e/ /d/ /I/ /n/ /b/ /E/ /r/ /E/
factory    = /f/ /ae/ /k/ /t/ /r/ /I/
            & = /f/ /ae/ /k/ /t/ /E/ /r/ /I/

```

The graphemes 'el' 'le' and 'en' also presented some problems. In cases where they were clearly pronounced with a schwa the vowel phoneme was coded separately, e.g.

```

kitchen = /k/ /I/ /ch/ /E/ /n/
pixel   = /p/ /I/ /ks/ /E/ /l/

```

In other cases the representation was not clear and the word was coded with and without the schwa:

```

jewels = /dz/ /ju/ /w/ /E/ /l/ /z/
        & = /dz/ /ju/ /w/ /l/ /z/
open    = /EU/ /p/ /n/
        & = /EU/ /p/ /E/ /n/
simple   = /s/ /I/ /m/ /p/ /l/
        & = /s/ /I/ /m/ /p/ /E/ /l/

```

In coding some other words the schwa was omitted:

```

people = /p/ /i/ /p/ /l/
little = /l/ /I/ /t/ /l/

```

The graphemes 'lm' 'ld' and 'lk' also present coding difficulties:

```

calmly = /k/ /a:/ /m/ /l/ /I/
milk    = /m/ /I/ /l/ /k/
could    = /k/ /u/ /d/
shoulder = /sh/ /EU/ /l/ /d/ /E/
          & = /sh/ /EU/ /l/ /d/ /E/ /r/
          & = /sh/ /EU/ /d/ /E/
          & = /sh/ /EU/ /d/ /E/ /r/
talk     = /t/ /o:/ /k/
          & = /t/ /o:/ /l/ /k/

```

The coding here is not consistent. The difficulties that it presents are discussed in chapter 8. Other ambiguities and problems with the coding are also discussed in that chapter.

### 7.3.7. Detail of the phoncode program

The phonetic coding program takes the input word, *inpw*, and segments it into all combinations of graphemes. The phonemes that can be represented by each grapheme are found by table look-up. All sequences of phonemes that may be considered to represent the *inpw* are compared with the dictionary. The dictionary is represented as a tree, where each node represents a phoneme, and also stores information about actual words (see figure 7-19). If a sequence of phonemes (representing *inpw*) matches a path in the tree, and the final node in the path also contains a word (stored as a string), then that word is offered as a phonetic match to *inpw*.

### 7.3.8. Segmenting the word

Each grapheme in the grapheme-phoneme table is compared with the *inpw* string. If there is a match, the phonemes corresponding to the grapheme, and the position of the grapheme in the word, are noted. If the character is marked as a 'tied' grapheme the characters in it are segmented no further: these characters in the word are no longer accessible. If the grapheme is not tied it may be segmented further. The next grapheme is compared with those remaining in the *inpw* string that are still accessible. Again, if the match succeeds, corresponding phonemes and position in the string are noted: tied phonemes are marked as inaccessible. The process is continued until all graphemes in the table have been compared, or until there are no accessible characters left in the string.

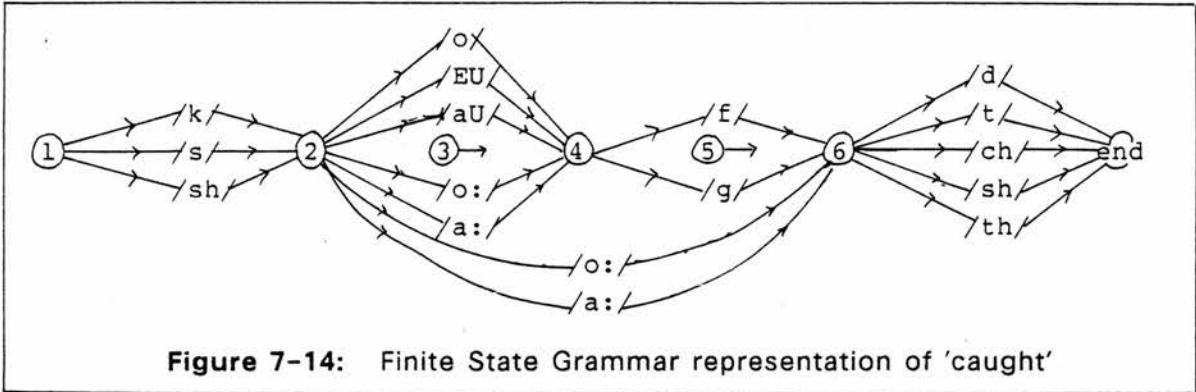
For example, if the *inpw* were 'caught', the grapheme 'augh' would match to characters 2 to 5 in the *inpw* string. The phonemes corresponding to this grapheme are /o:/ and /a:/. The grapheme 'augh' is not tied, and may be further split. The next matching grapheme in the table is 'gh', with phonemes /f/ and /g/. This grapheme is marked as tied, so no further segmentation of the grapheme 'gh' is required. These matches may be represented in the following format:

```
grapheme => alternative phonemes
augh => /o:/ /a:/
gh    => /f/ /g/
```

The remaining matches found are:

au => /EU/ /aU/ /o:/ /a:/ /o/  
 c => /k/ /s/ /sh/  
 t => /t/ /ch/ /sh/ /d/ /th/

The set of phonemes representing the inpw may also be represented as a finite state grammar, as in figure 7-14.



If  $n$  is the length of the inpw, there will be  $(n+1)$  nodes, the  $(n+1)$ th being the end node. Each node indicates the position of a character in the inpw: arcs leaving the nodes represent phonemes corresponding to the graphemes commencing with that character, in that position. Some nodes have no arc connected to them: these represent the characters occurring in tied graphemes. In figure 7-14 the character 'h' in the string 'caught' corresponds to node 5. The grapheme 'gh' is marked as tied, represented by phonemes /f/ and /g/. The arcs leaving the 'g' node, node 4, are labelled /f/ and /g/. No arc goes to or from the 'h' node, node 5: it has no corresponding phoneme in this case.

The representation of the phonemes corresponding to a word may also be given in a slightly different form, as in figure 7-15.

In this format, the nodes are numbered, and the phonemes labelling arcs from the node are given, as before. The arcs are replaced by explicit values of the 'next node', from each node. In the finite state grammar representation, a sequence of phonemes representing the string is found by traversing the arcs in the direction indicated from node 1 to the end node, selecting any one arc from each node, and moving to each node connected by the arcs, in turn. The labels of each arc traversed are the phonemes representing the string. All possible sequences are found by following each arc from a node, in order, and backtracking until all paths between node 1 and the end node have been

```

inpw='caught'
node = 1
    phon = /k/ , next = 2
    phon = /s/ , next = 2
    phon = /sh/ , next = 2
node = 2
    phon = /o:/ , next = 6
    phon = /a:/ , next = 6
    phon = /O/ , next = 4
    phon = /EU/ , next = 4
    phon = /aU/ , next = 4
    phon = /o:/ , next = 4
    phon = /a:/ , next = 4
node = 3
node = 4
    phon = /f/ , next = 6
    phon = /g/ , next = 6
node = 5
node = 6
    phon = /d/ , next = end
    phon = /t/ , next = end
    phon = /ch/ , next = end
    phon = /sh/ , next = end
    phon = /th/ , next = end
node = end

```

**Figure 7-15:** Alternative representation of 'caught'

traversed. In the alternative representation, a sequence of phonemes is found by starting at node 1, selecting any phoneme from the list for that node, and moving to the 'next node' as indicated. Further phonemes are selected, and moves to the 'next node' made, until the end node is encountered. All possible sequences are found by selection of each phoneme from each node list in turn, backtracking to cover all possible combinations.

We might consider listing all possible sequences of phonemes representing the *inpw*: however, even for this simple example, there are 180 possible sequences, or paths. This is calculated as follows. If the number of possible paths between nodes 1 and 2 is rewritten as  $\text{paths}(1,2)$ , and the path from node 1 to end via nodes 2,4, and 6 is written as  $[1,2,4,6,\text{end}]$  :

```

paths(1,2)    = 3
paths(2,4)    = 5
paths(2,6)    = 2
paths(4,6)    = 2
paths(6,end)  = 5

```

```

total paths(1,end) = (3 x 5 x 2 x 5) [1,2,4,6,end]
                   + (3 x 2 x 5)    [1,2,6,end]
                   = 150 + 30
                   = 180 possible paths

```

The set of graphemes and corresponding phonemes, used in the phoncode program is given in figure B-1 in appendix B. A "1" preceding a grapheme indicates that it is tied; a "0" indicates that it may be segmented further; a "\*" represents the 'vc+e' construction. So, in the word 'cane' the 'a\_e' is represented in the grammar by '\*a'. Similarly, in 'raise', 'ai\_e' is represented by '\*ai'. The \* marked graphemes are matched if the vowel or vowel digraph matches, and the next but one character after it is 'e'. There are two sets of representation for the 'vc + e' grapheme. In the first, the segmentation is 'v' and 'c+e': the 'e' is not represented by any separate phoneme. In the finite state grammar representation, the arc representing the consonant preceding the 'e' passes directly to the node corresponding to the character following the 'e'. For example, the grapheme-phoneme correspondences for 'rime' are:

```

i_e => /i/ /aI/ /I/ /e/ /E/
m    => /m/
r    => /r/ /e:/ /E/

```

In the second representation of 'vc+e', the structure is split 'v' and 'c' and 'e', the vowel or vowel digraph, the consonant, and the 'e' all being represented independently:

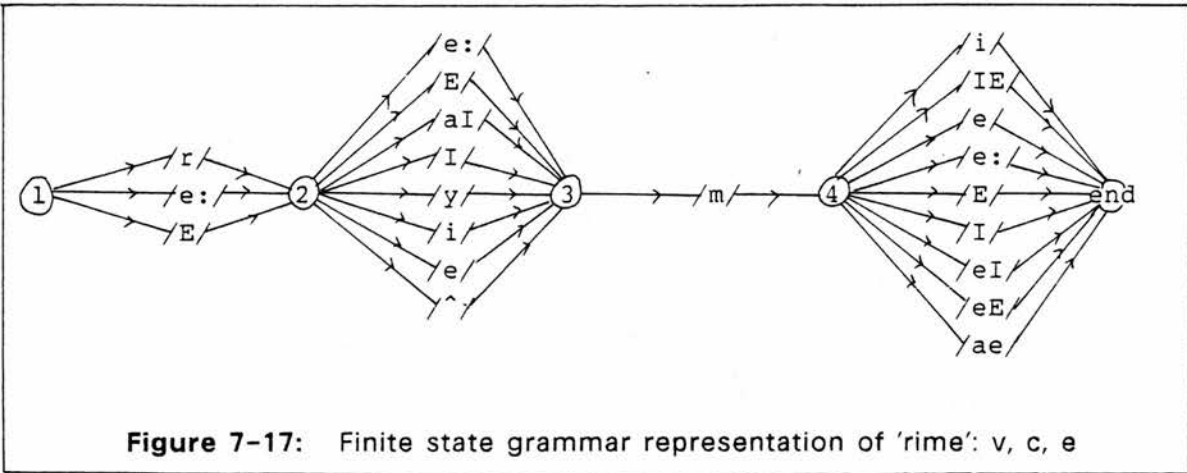
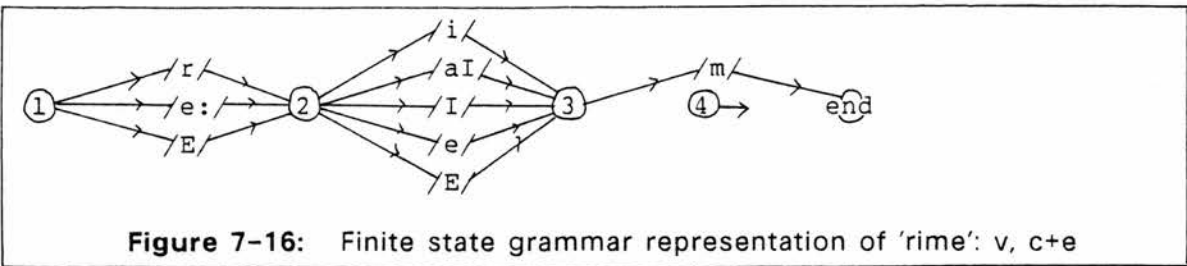
```

e => /i/ /IE/ /e/ /e:/ /E/ /I/ /eI/ /eE/ /ae/
i => /e:/ /E/ /aI/ /I/ /y/ /i/ /e/ /~/
m => /m/
r => /r/ /e:/ /E/

```

The corresponding finite state grammars for each are shown in figure 7-16 and figure 7-17.

These two sets of phoneme representation for 'rime' cannot be considered in one finite state grammar form: the arcs between nodes 2 and 3 in the first case ( with 4 passed over) are not alternatives for the arcs between nodes 2 and 3



in the second case. There is no legitimate path following the arcs between nodes 2 and 3 in the second case, along arc labelled /aI/, and then following arcs between nodes 3 and the end in the first case, along arc labelled /m/.

The two sets of representation for 'vc+e' graphemes can, however, be combined in the 'alternative representation' format. In figure 7-18 the full representation of 'rime' is given. To indicate the form in which 'e' is passed over, the 'next node' of the vowel has a value of the corresponding (following) consonant node, plus 100. If a vowel phoneme is selected, with a nextnode value greater than 100, the node for the consonant is taken (i.e. the next node less 100) and a phoneme value is obtained: the value of the consonant nextnode is increased by 1, and this new value becomes the following node, i.e. the 'e' node is passed over. In this example if at node 2 the phoneme /<sup>^</sup>/ were selected (next=103), then at node 3 the phoneme /m/, next=end, would be chosen, and any arcs from node 4 would be skipped over.

Further examples of 'parses' of input words are given in appendix B.

```

inpw = 'rime'
node = 1
    phon = /r/      next = 2
    phon = /e:/     next = 2
    phon = /E/      next = 2
node = 2
    phon = /i/      next = 3
    phon = /aI/     next = 3
    phon = /I/      next = 3
    phon = /e/      next = 3
    phon = /E/      next = 3
    phon = /e:/     next = 3
    phon = /^/      next = 3
    phon = /y/      next = 3
    phon = /E/      next = 103
    phon = /aI/     next = 103
    phon = /I/      next = 103
    phon = /i/      next = 103
    phon = /e/      next = 103
node = 3
    phon = /m/      next = 4
    phon = /m/      next = end
node = 4
    phon = /i/      next = end
    phon = /IE/     next = end
    phon = /e/      next = end
    phon = /e:/     next = end
    phon = /E/      next = end
    phon = /I/      next = end
    phon = /eI/     next = end
    phon = /eE/     next = end
    phon = /ae/     next = end
node = end

```

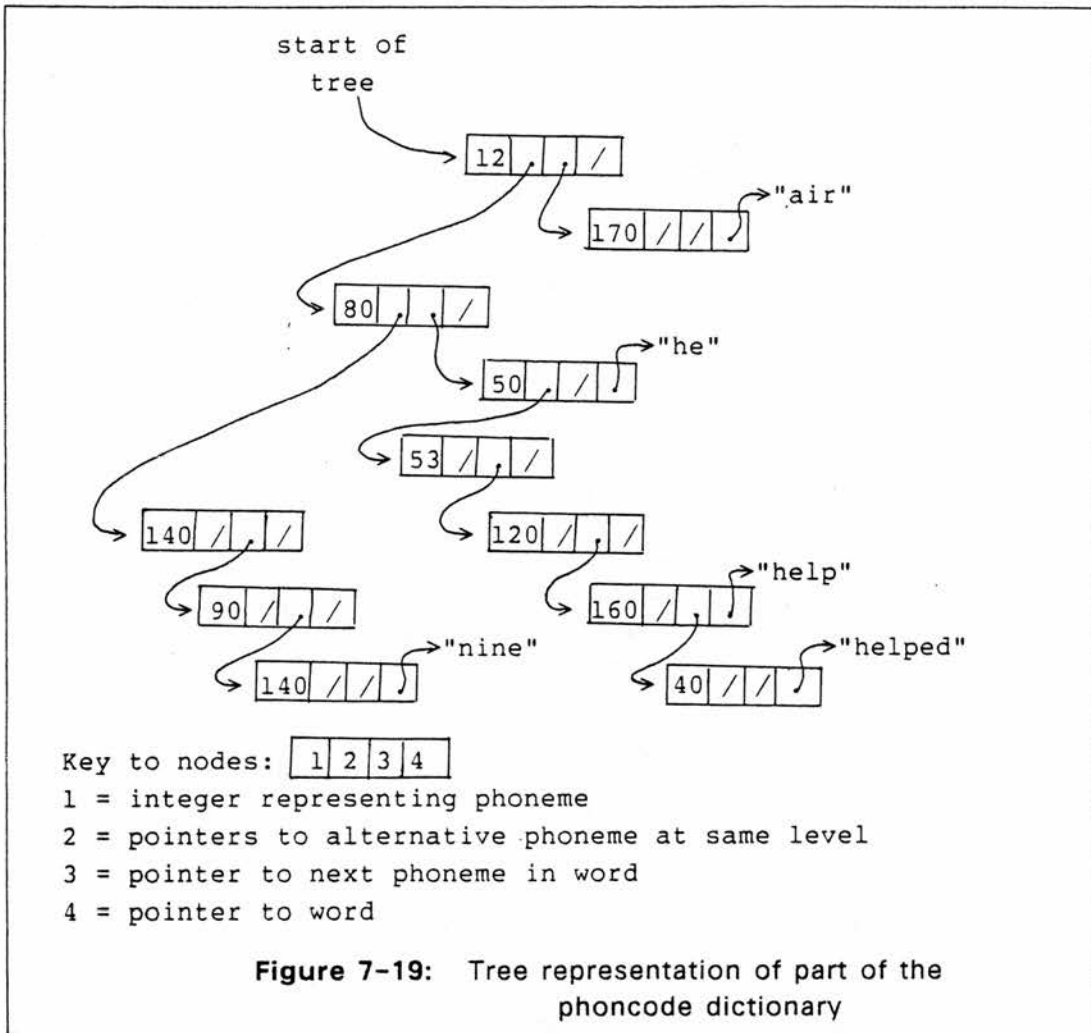
**Figure 7-18:** Representation of 'vc + e' graphemes

### 7.3.9. Representing the dictionary

Given a file of words coded phonemically, a dictionary is constructed representing the set of words as a tree where each node stores a phoneme value. The phoneme representation used is the integer one, given in figures 7-12 and 7-13. Examples of words and their phonemic representation are:

he	/h/ /i/	80 50
help	/h/ /e/ /l/ /p/	80 53 120 160
helped	/h/ /e/ /l/ /p/ /d/	80 53 120 160 40
nine	/n/ /ai/ /n/	140 90 140

These are represented in tree form in figure 7-19.



The dictionary tree to represent these five words would be built up as follows. The first phoneme in each of the words 'he', 'help' and 'helped' is the same, 80, and so would all be stored in the same node. The second phoneme is 50 in 'he' and 53 in 'help' and 'helped'. The right offspring of the node storing 80 will store 50: the left offspring of this node (50) stores the 'alternative phoneme', 53. For other words with first phoneme 80 and an alternative second phoneme, e.g. 'hill' 80 93 120, the second phoneme is added in the left offspring of the node storing 53. The word 'he' is represented by two phonemes, both now stored: the string 'he' is also stored in the node with the last phoneme in the word, 50. The third phoneme in 'help', 120, is stored in the right offspring of that with 53. This also matches the next phoneme in 'helped' and does not need to be represented twice. The right offspring of this node (storing 120)

stores the fourth phoneme, 160. This is the last phoneme in 'help', so the string 'help' is also stored in this node. The final phoneme in 'helped' is added as the right offspring, storing 40 and the string 'helped'.

The word 'nine' has first phoneme 140, and is stored as the left offspring (alternative first phoneme) to the node containing 80. The remaining phonemes in the word 'nine' are stored in the right offspring of this node (storing 140). The string 'nine' is stored also with the final phoneme.

The word 'air' is coded 12 170. It does not match 80 or 140, the alternative initial phonemes. It must be added as another alternative initial phoneme, as a left offspring. However, the left offsprings are ordered in increasing order. The node storing 12 becomes the root node and the parent of the node storing 80, this latter node becoming its left offspring: the new node is inserted in the tree. The second phoneme is added as the right offspring of this new node and the string 'air' stored also.

Further words may be added. The initial phoneme is compared with each left offspring node, starting with the current root node. If no match is found the first phoneme of the word is inserted in the correct position in the ordering; the remaining phonemes in the word are added as right descendants. If a match is found the next phoneme (the second) is compared firstly with the right offspring of the matched phoneme node, and then with its offsprings, and left descendants of this offspring. Again, if no match is found, a new node is inserted in the correct position, and the remaining phonemes added; otherwise, if a match is found, the next phoneme is compared with firstly the right offspring and then with its left descendants. The process is repeated until all phonemes in the word are matched, or have been added/inserted. The word string is stored in the same node as the last phoneme in the word. In this way a file of phonemically coded words may be stored as a tree structured dictionary.

### 7.3.10. Matching the word and dictionary

The comparison part of the phoncode program takes the dictionary of phonemically coded words, represented as a tree, and an inpw, parsed to show all sequences of phonemes that could have produced it. The alternative representation of the parsed inpw (as in figure 7-15) will be used here to describe the matching process: it will be referred to as the inpw network. In the program the integer phonemic coding is used,

Starting at the root of the dictionary tree, and the first node of the inpw network, the first phoneme listed at the inpw node 1 is compared with the dictionary root node phoneme. The phonemes listed at the inpw nodes are in increasing order, as are the left descendant dictionary nodes. If the compared phonemes match, then the next inpw node is found (numbered as 'next') and the first phoneme of this new inpw node is compared with the right offspring phoneme of the matched dictionary node. If the match succeeds then the comparison process is repeated. If the match fails, the inpw node phoneme is compared with the left offspring dictionary node phoneme. Taking account of the fact that dictionary and inpw nodes are ordered, whenever a match fails the next inpw node or dictionary word node is tried. For example, when an inpw phoneme 80 fails to match a dictionary node phoneme 60, and also fails to match its left offspring 90, it is pointless to compare it with the left descendants of the node storing 90, as the phonemes will be greater than 90. Instead the next inpw phoneme is compared with the dictionary node storing 90. If this inpw phoneme is 120 then the match would again fail, and the next left offspring of the dictionary would be compared instead. If at any point there are no more phonemes listed at the inpw node, or no more left offspring phonemes to compare, then the program backtracks to the last pair of nodes matched, fails that match, and continues. If matching succeeds such that the end node of the inpw network is reached then any word strings stored in the last matched dictionary node are saved as 'possible phonemic equivalents' for the inpw. All paths through the inpw network are tried and the dictionary is searched exhaustively. All possible matches are found.

In appendix B a set of examples is given: the phonetic coding of a small example dictionary is shown; examples of inpws, parsed to produce all sets of

phonemes, are given. Successful matches for each inpw, from the example dictionary, are given.

The editcost and phoncode programs described in this chapter were tested on a corpus of errors produced by children with spelling disabilities. Additionally, the editcost program was used and tested by a small group of children (see Study 2 in chapter 6). The performance of each of the programs is discussed in chapter 8.

## Chapter 8

### Performance

#### 8.1. Introduction

In this chapter the performance of the editcost and phoncode programs, in correcting the errors made by the children, is assessed.

If the programs are able to provide correction of the errors, then this provides evidence that:

- (a) there are regularities in the children's errors;
- (b) information relating to these regularities can be used by the programs to reconstruct the correction from the error.

Where there is failure to correct an error, this can be attributed to one or more of the following:

- (a) the errors are not sufficiently regular;
- (b) the programs do not have sufficient information about the regularities of the errors i.e. the grammar or weightings are incomplete or incorrect;
- (c) the algorithm fails: sufficient information about existing regularities may be supplied to the program, but still there is failure to reconstruct the error.

Relating to these possible sources of error, the following question is also considered:

- \* Is a human judge able to perceive regularities in the errors, and would he/she then be able to provide corrections?

The editcost and phoncode programs are each considered in relation to the following questions:

1. Does the program succeed in correcting the errors made?

2. If there is a failure, is it due to:

(a) the errors being irregular,

(b) the program data being insufficient or incorrect,

(c) the methods of analysis being unsuitable?

3. When the programs succeed, what does this tell us about

(a) the individual children

(b) the methods of correction?

## 8.2. Performance of the editcost program

The performance of the editcost program was initially assessed on two sets of data:

1. The words used with the editcost program in study 2 (S2);
2. The complete set of errors made in studies 1 and 2 (S1, S2).

### 8.2.1. Testing editcost in use - Study 2

The editcost program was used in study 2 (S2), as described in chapter 6. Each child used the program whenever he wished to check the spelling of a word (i.e. the input word). In some cases the word that was checked was correctly spelt: in other cases it was misspelt. It was compared with the set of words shortlisted from the dictionary. The dictionary consisted of the words in the generaldict, and the topic dictionary words for the particular session. The four words with lowest minimum repair cost were found and offered as possible corrections.

Whenever a word was checked, the outcome could be categorised in one of three ways:

- (i) the correction for the input word was both in the dictionary and offered as a possible correction;
- (ii) the correction was in the dictionary, but was not offered as a possible correction;
- (iii) the correction was not in the dictionary.

The frequency of occurrence for each category i, ii, iii, for each group of children taking part in S2 is given in figure 8-1. Group 1 comprises FR, DV, TE and DR; group 2, DI, MA, GR and ST.

These results can be re-expressed as percentages.

Percentage correction offered, overall:  $i/\text{total}$

group 1	group 2	both groups
73.6%	64.7%	71.2%

Percentage correction offered, when in the dictionary:

	i correction in dictionary and offered	ii correction in dictionary not offered	iii correction not in the dictionary	iv total
Group 1	229	27	55	311
Group 2	77	3	39	119
Both groups	306	30	94	430

**Figure 8-1:** Editcost in use: outcomes of checking

$i/(i+ii)$

group 1	group 2	both groups
89.5%	96.3%	91.1%

and from this, percentage correction **not** offered when  
in the dictionary:  $ii/(i+ii)$

group 1	group 2	both groups
10.5%	3.7%	8.9%

Percentage of corrections not in the dictionary:  $iii/iv$

group 1	group 2	both groups
17.7%	32.8%	21.9%

From these results it can be seen that the program was able to offer the correction for a large percentage (>90%) of the words checked, assuming that they were in the dictionary. The correction algorithm was more successful for group 2 than for group 1 (96% vs. 89% corrections). However, group 2 attempted to check the spelling of the larger percentage of words that were not in the dictionary.

These results may also be considered for individual children, as in figures 8-2 and 8-3.

Results for groups 1 and 2 were compared using the Mann-Whitney U test (one-tailed). This test was also used to assess differences in performance of groups 1 and 2 in the first study, and for all other group comparisons in this chapter.

	i . number corrected	i/iv % of the total	i/(i+ii) % of those in the dictionary	iv total number checked
Group 1				
FR	69	71.9%	92%	96
DV	50	82%	87.7%	61
TE	72	74.2%	90%	97
DR	38	67.7%	86.4%	57
Group 2				
DI	10	66.7%	100%	15
ST	20	62.5%	100%	32
MA	27	65.9%	96.4%	41
GR	20	64.5%	90.9%	31
Total	306			430

**Figure 8-2:** Editcost in use: individual results  
Correction offered

	ii		iii		iv
	correction in the dictionary		correction not in the dictionary		total number
	freq	% of total	freq	% of total	checked
Group 1					
FR	6	6.3%	21	21.9%	96
DV	7	11.5%	4	6.6%	61
TE	8	8.2%	17	17.5%	97
DR	6	10.5%	13	22.8%	57
Group 2					
DI	-	0%	5	33.3%	15
ST	-	0%	12	37.5%	32
MA	1	2.4%	13	31.7%	41
GR	2	6.5%	9	29%	31
Total	30		94		430

**Figure 8-3:** Editcost in use: individual results  
Correction not offered

From figure 8-2 it can be seen that group 2 showed a higher percentage correction of words in the dictionary than group 1 ( $p < 0.05$ ). The number of corrections offered, taken as a percentage of the total number of words checked, was higher for group 1 ( $p < 0.02$ ). Within groups there is little difference shown in the percentage of errors corrected (the range being  $< 10\%$ ). Between groups there is less than 15% difference between the highest (100%

for ST and DI) and the lowest (86.4% for DR). For the majority of cases where the correction was not offered, (c. 75%) the correction was not in the dictionary (see figure 8-3). The exception to this was errors made by DV – however, he showed the highest correction success rate overall. Group 1 had a significantly greater percentage of errors that were in the dictionary and not corrected ( $p < 0.02$ ) than group 2; group 2 showed a greater percentage of errors for which the correction was not in the dictionary ( $p < 0.02$ ).

The possible corrections were ordered by cost, the lowest cost being offered first. The intended word, if it was included in the possible corrections, could be the first word offered (off(1)) or in the second, third or fourth positions (off(2/3/4)). The corrections offered were categorised according to whether they were off(1) or off(2/3/4). For each group the percentage of first words offered was:

	Group 1	Group 2	Both groups
off(1)	76.4%	97.4%	81.7%
off(2/3/4)	23.6%	2.6%	18.3%

For group 1 the intended correction was the first word offered in three-quarters of cases. For group 2 it was off(1) in more than 97% of cases. Overall, in more than four-fifths of cases the intended correction was offered as the possible correction with the least cost repair.

On a number of occasions, if a word was checked and the correction not offered the child was encouraged to re-check it with a different spelling: "One closer to the correct word". These rechecks are included in the above categories, according to their outcomes. For group 1, 21 of the 27 words (in category ii) were rechecked with a different spelling. For 19 of these, the correction was found and offered. For group 2, for all 3 words in category ii, the correction was offered when rechecked. So, for the combined groups, of the 30 words for which the correction was not offered, 24 were rechecked with different spellings, 22 of these rechecked words produced the required spelling.

When the required word was not in the dictionary, the investigator could be asked to add it. The initial spelling could then be rechecked. Twenty-six words were added and rechecked, 11 from group 1 and 15 from group 2. With the exception of 1 word from group 2, the corrections were offered for all added and rechecked words.

The words that were not corrected successfully by the algorithm are discussed in more detail at the end of this section, in subsection 8.2.3.

### 8.2.2. Testing on the corpus containing Study 1 and Study 2 errors

The editcost program was tested on the corpus of errors made by the children in both studies. These errors included those checked with the editcost program (S2), those made when writing (S2), and those made when typing (S1 and S2). Chapter 6 gives details of the two studies. The dictionary used was set up specifically for testing. Whilst the dictionary that had been used in each of the S2 sessions contained 750 to 1000 words, the testing dictionary contained more than 2000 words. It comprised the general dictionary, plus all the topic dictionaries and all the corrections of errors (with duplicates removed).

Each error was checked using the editcost program, and the five dictionary words with lowest minimum cost repair were recorded. The reason for recording the fifth word was to test whether the performance would be substantially improved if it was included in the possible corrections. In only two cases in S1 and 15 cases in S2 was the correction the fifth option. This represents 2% of the total number of errors. In these results the correction offered as the fifth option is not counted as a success.

For each child the following information was recorded:

- (a) the number of errors for which the correction was offered;
- (b) the percentage of errors for which the correction was offered;
- (c) the number of the corrections that were offered as first option (off(1));
- (d) the percentage of the corrections that were offered as first option;
- (e) the total number of errors made.

The results of testing the errors made in S1 are given in figure 8-4.

The program offered the correction for 85% of errors, over both groups. 93.2% of errors made by group 1 were corrected, whilst 80.3% of corrections were offered for group 2 (not significant). It was least successful for CM and SS,

	a. number corrected	b. % corrected	c. number off(1)	d. % of a. off(1)	e. total number
Group 1					
GQ	15	100%	13	86.6%	15
JM	38	100%	33	86.8%	38
MN	29	82.9%	24	82.8%	35
Group 2					
LB	24	92.3%	22	91.7%	26
NM	30	90.9%	28	93.3%	33
CM	46	76.7%	34	73.9%	60
SS	18	64.3%	14	77.8%	28
Group 1 total	82	93.2%	70	85.4%	88
Group 2 total	118	80.3%	98	83.1%	147
Both groups	200	85.1%	168	84%	235

**Figure 8-4:** Editcost tested on Study 1 errors

offering only 64% of corrections in the case of SS (the reasons for this failure are discussed in section 8.5). It was most successful for GQ and JM, providing 100% correction. In 84% of cases where the correction was offered it was the first option i.e. it had lowest edit cost. Note that the program weightings were based on the frequency of errors made by this group, and therefore a high percentage of corrections offered was to be expected.

The results of testing the errors made in S2 are given in figure 8-5.

The same information is given, as for S1. Corrections were offered for nearly 79% of errors, over both groups. The first option offered was the correction in 73% of errors overall.

### 8.2.3. Errors which the editcost program failed to correct

The errors for which the editcost program did not offer corrections will now be considered, and reasons for this failure discussed. The sets of errors on which the program failed are given in figures 8-6 (use by S2), 8-7 (testing on S1 errors), 8-8 (testing on S2, group 1 errors) and 8-9 (testing on S2, group 2 errors).

	a. number corrected	b. % corrected	c. number off(1)	d. % of a. off(1)	e. total number
Group 1					
FR	103	83.7%	81	78.6%	123
DV	65	66.3%	39	60%	98
TE	106	80.9%	66	62.3%	131
DR	55	63.2%	41	74.5%	87
Group 2					
GR	45	81.2%	34	61.8%	55
DI	21	95.5%	19	86.4%	22
MA	39	92.9%	33	78.6%	42
ST	39	92.9%	33	78.6%	42
Group 1 total	329	74.9%	227	69%	439
Group 2 total	144	89.4%	119	82.6%	161
Both groups	473	78.8%	346	73.2%	600

**Figure 8-5:** Editcost tested on Study 2 errors

An error for which the editcost program does not offer the correction will be referred to as a **non-corrected error**. The set of non-corrected errors resulting from the use of the program in S2 is a subset of those resulting from testing all the S2 errors, and so this subset will not be considered separately.

Non-correction of an error indicates the inability of the program to reconstruct the correction from the error. This could be due to:

1. errors being so irregular that the correction cannot be inferred;
2. program data being incomplete or incorrect, that is:
  - (a) omission of the correction in the shortlisting process;
  - (b) the weightings used being inappropriate;
  - (c) the costing function being inappropriate;
3. the description of the errors in terms of format (and hence analysis in terms of edit operations) being inadequate.

The latter two possible causes of failure will be considered first.

Inclusion of a dictionary word in the shortlist, for consideration by the costing algorithm, was dependent upon the length and first character(s) of the word. In a number of cases, the desired correction was omitted from the shortlist.

Non-correction of the misspelling is attributable to a failure in shortlisting for:

9 out of 35 non-corrected errors in S1  
 24 out of 127 non-corrected errors in S2  
 33 out of 162 non-corrected errors in total

If further alternatives are permitted for first letter confusions, more corrections could be included in the shortlist. For example, alternatives a for u (=a/u), e/i, g/b, wh/ho, t/ch, h/th would reduce the omissions from the shortlist by 6. Additionally, if a difference of 4 characters is permitted between word and error, for words of less than 10 characters, then a further 5 words would be shortlisted.

However, the program does succeed in providing the correction for 85.1% of S1 errors and for 78.8% of S2 errors when tested, and for 91.1% of S2 errors checked (for which the correction is available) when the program is in use (see section 8.2.1): more than 80% of errors tested overall. For a large number of

FR		DV	
eyes	irs	brown	blounm
eyes	ias	hair	hear
head	hard	of	ove
saw	sore	buried	beray
saw	sour	of	ovre
about	ubout	dalglish	dugle
		magazine	magen
TE		DR	
gold	goib	strachan	stacking
through	thr	strachan	cracking
through	thro	instructions	inchuns
called	golld	instructions	chuns
bunny	bune	turtle	trener
any	ene	turtle	turend
conservative	cunjnc		
conservative	sevter		
MA		GR	
		computer	ucnputer
perq	pirck	computer	uncomputer
DI and ST - no uncorrected errors			

**Figure 8-6: Using editcost - Study 2**  
Errors for which correction not offered

MW		NM	
won	one	paw	po
threw	through	change	caing
a	are	we	wer
change	gh	CM	
wrote	nrote	called	colde
wrote	krote	commercial	commrs
		university	ynusty
LB		can	came
quarry	qorie	draw	droy
fool	full	that	ther
SS		recall	tecall
new	neea	change	calde
had	hat	night	nairt
draw	john	the	whe
make	mosea	haunted	hoted
make	msea	through	thro
time	the	came	cane
get	cedt	hear	haes
television	tahgfring		
tv	talhfi		
horror	horey		
GQ and JM - no uncorrected errors			

**Figure 8-7: Testing editcost - Study 1**  
Errors for which correction not offered

## Group 1

FR		TE	
hair	hera, hare	weight	wait
eyes	irs, ias	gold	goib
head	hard, herd	through	thro, thr
silver	isilver	came	gam
saw	sore, sour	called	golld
where	warh	come	conn
of	fo	seen	cn, cen
piece	pees, peces, peesc	motor	moterdf
about	ubout	dangerously	bandrie
showed	sods, shodes, sodes	bunny	bune
wyse	wizes	conservative	cunjnc, sefter
put	pit	plastic	plasek
		of	over
		work	wrk
		any	ane
		would	wob
		we	wie
		had	thad
		just	tust
		walked	workt
		took	tike
		dark	barck
DV		DR	
eyes	liss, isse	have	haft, half
brown	bloum, blounm	light	like
hair	hear	the	then
bye	bi	check	pellrs
island	illing	packed	park, par, part,
of	over, ove, ov, ovre	won	parck, pakt
buried	beray	strachan	win
ghost	goss		stacking,
dalglish	dugle	nicholas	cracking
have	uve	turtle	nickris, nickis
for	of		turned, trener
goodbye	boodbye	stadium	turend
magazine	magen	brazil	stamun
interview	intovue	picture	brasur
soldiers	soildde	instructions	ping
plastic	plaiked		inchins, incruns
stairs	stared, stare	robot	chuns, inchuns
down	dame		romdt, rodet,
pictures	pieces		rodert, roder,
could	cood		romdert
bit	bid	dead	beb
talk	tock	dog	bog
drill	drule		
more	mor		
photo	fot, front		
white	withe		

**Figure 8-8:** Testing editcost - Study 2, group 1  
Errors for which correction not offered

Group 2			
GR		DI	
called	could	specifications	spec
straight	strat		
try	trie	MA	
who	how	perq	purk, pirck
uses	yous	alternatives	alteration
a	and		
put	pit	ST	
any	ena	tune	chune
computer	uncomputer,	procedures	prgame
	ucnputer	so	sow

**Figure 8-9:** Testing editcost - Study 2, group 2  
Errors for which correction not offered

errors, therefore, it seems that their description in terms of format, assignment of weightings and calculation of costs, is sufficient to enable reconstruction.

It may be that some of the spellings are so bizarre that they conform to no apparent pattern: the correction will not be recognizable from the error. To test this, the set of non-corrected errors (for S1 and S2) was given to an independent judge for correction. The judge was asked to write what he thought would be the correction for each misspelling alongside it; to mark with a tick any word that he thought was spelt correctly (i.e. words misspelt as other words would be marked); to mark with a cross any word for which he could suggest no correction. Having corrected or marked all words presented, the judge was then told that, in fact, all the words were misspellings. He was then asked to write alongside each ticked word (apparently correct words) what he thought the spelling could be (knowing that it was not the word given). The judge's corrections were then compared with the intended corrections, and all discrepancies noted.

If the judge had succeeded in correcting all the errors, where the editcost program failed, this would suggest that improvements of the program were needed. On the other hand if the judge failed to correct the majority of errors (i.e. they were unrecognizable) then this would indicate too little consistency, or lack of identifiable pattern, in the errors made. That more than 80% of errors were successfully corrected, by the program, indicates that there is an identifiable pattern in the majority of errors.

It might be argued that the judge might fail to correct the errors because of unfamiliarity with the vocabulary used by the children in the two studies. This was overcome by using the same judge who had already seen all sets of error-correction pairs (see subsection 8.3.2). This meant that the judge had seen all the errors before, with their corrections, though in a different order (errors were presented in a random order). He was also reminded of the topics dealt with in the children's writing. Despite this, he failed to recognize a substantial number of errors, though he did indicate that his previous experience had slightly influenced the corrections offered.

Outcomes of comparison of the judge's corrections and the intended corrections are classed as follows:

1. the correction provided by the judge was the intended word (=C)
2. no correction could be suggested (=NC)
3. the wrong correction was suggested (=WC)
4. the misspelling was taken as the correct spelling of another word initially, but was later reconsidered and classed in one of the above categories (=IC, INC, IWC)

A summary of the results is given, for each group, in figure 8-10. The total frequencies for the categories C, NC, and WC are given. Included are those errors initially thought to be correct (the frequency of which are given in brackets, for each category).

	Correction C	No Correction NC	Wrong Correction WC	Total
Study 1				
Group 1	2(1)	2(0)	2(2)	6(3)
Group 2	5(2)	9(1)	15(1)	29(4)
S1 total	7(3)	11(1)	17(3)	35(7)
Study 2				
Group 1	29(6)	38(11)	43(7)	110(24)
Group 2	12(2)	3(3)	2(0)	17(5)
S2 total	41(8)	41(14)	45(7)	127(29)
Total	48(11)	52(15)	62(10)	162(36)

**Figure 8-10:** Comparison of judge's corrections with intended corrections - summary

Results are also given, for each child, in figures 8-11 and 8-12.

	C	IC	NC	INC	WC	INC	Total
Group 1							
GQ	-	-	-	-	-	-	0
JM	-	-	-	-	-	-	0
MW	1	1	2	-	-	2	6
Group 2							
LB	-	1	-	-	1	-	2
NM	-	-	1	-	2	-	3
CM	3	-	3	-	7	1	14
SS	-	1	4	1	4	-	10
Group 1 1	1	1	2	0	0	2	6
total							
Group 2 3	2	2	8	1	14	1	29
total							
Both	4	3	10	1	14	3	35
groups							

**Figure 8-11:** Comparison of judge's corrections with intended corrections - Study 1

	C	IC	NC	INC	WC	INC	Total
Group 1							
FR	7	2	3	-	4	4	20
DV	7	1	5	5	12	3	33
TE	6	1	10	1	7	-	25
DR	3	2	9	5	13	-	32
Group 2							
GR	5	2	-	2	1	-	10
DI	1	-	-	-	-	-	1
MA	2	-	-	1	-	-	3
ST	2	-	-	-	1	-	3
Group 1 23	6	6	27	11	36	7	110
total							
Group 2 10	2	2	0	3	2	0	17
total							
Both	33	8	27	14	38	7	127
groups							

**Figure 8-12:** Comparison of judge's corrections with intended corrections - Study 2

The judge corrected 29.6% of the non-corrected errors. He failed to offer a correction for 32.1% of the errors and offered alternatives for 38.3%. Of those corrected 11 had initially been believed to be alternative words, spelt correctly, and were left uncorrected. At first attempt, therefore, only 22.8% of errors were successfully corrected. Overall the judge failed to correct 70.4% of errors. Thus for 70.4% of errors that the program failed to correct, the human judge also failed to identify the correction; despite knowing that all the words were errors and having previously seen the error/correction pairs. Additionally, of the 33 words that the program failed to shortlist, 20 presented difficulty to the judge. The judge experienced particular difficulty with the errors made by CM and SS (S1, group 2) and by DV, TE and DR (S2, group 2): he failed to correct between 72% and 90% of them. This suggests that these errors were in some way unrecognizable.

Summarising the results for the editcost program overall:

- \* the program succeeded in correcting

- (a) 85.1% of errors made by Group 1, when tested;
- (b) 78.8% of errors made by Group 2, when tested;
- (c) 91.1% of errors made by Group 2 (for which the correction was available) when the program was in use;
- (d) 80.6% of errors tested (a + b) overall.

- \* of those it failed to correct (162 errors)

- (i) 48 were corrected by the judge (therefore attributable to failure on the part of the program), accounting for 5.7% of errors overall;
- (ii) 114 were not corrected by the judge (therefore attributable to insufficient regularities shown in the errors), accounting for 13.7% overall.

### 8.3. Performance of the phoncode program

#### 8.3.1. Testing on Study 1 and Study 2 errors

The performance of the phoncode program was assessed on the sets of errors made in S1 and S2. The same testing dictionary was used for testing both the phoncode and editcost programs. The dictionary was coded phonemically for testing with the phoncode program (see chapter 7, section 7.3.6).

Each error was input to the phoncode program. Words offered by the program as 'phonetic equivalents' were recorded. Whether or not the correction for the error was included in these words was noted. The following information was obtained:

- (a) the number and percentage of errors for which the correction is included in the words offered by the program;
- (b) the number and percentage of errors for which the correction is not offered;
- (c) the total number of errors made.

The results of testing the errors in S1 are given in figure 8-13.

	a. correction included in words offered		b. correction not included in words offered		c. total number of errors
	freq	%	freq	%	
Group 1					
GQ	10	66.7%	5	33.3%	15
JM	31	81.6%	7	18.4%	38
MN	21	60.0%	14	30.0%	35
Group 2					
LB	23	88.5%	3	11.5%	26
NM	21	63.7%	12	36.3%	33
CM	30	50.0%	30	50.0%	60
SS	8	28.6%	20	71.4%	28
Group 1	62	70.5%	26	29.5%	88
Group 2	82	55.8%	65	44.2%	147
1 & 2	144	61.3%	91	38.7%	235

**Figure 8-13:** Phoncode tested on study 1 errors

The percentage of errors for which the correction is included in the words offered, for all children, is 61.3%. The overall percentage for group 1 is higher than that for group 2, though the difference is not statistically significant. The lowest percentage offered is 28.6% for SS (more than 20% lower than for any other child). CM is next lowest with 50% corrected. MN, NM and GQ all fall in the 60 to 67% range. The highest percentage corrections are for LB and JM, with 88.5% and 81.6% respectively. Information is given for each child, and for each group of children. The results of testing errors made in S2 are given in figure 8-14. The same information is provided for this group.

	a.		b.		c.
	correction included in words offered		correction not included in words offered		total number of errors
	freq	%	freq	%	
Group 1					
FR	78	63.4%	55	36.6%	123
DV	43	43.8%	45	56.2%	98
TE	71	54.2%	40	45.8%	131
DR	30	34.5%	57	65.5%	87
Group 2					
GR	35	63.6%	20	34.4%	55
DI	16	72.7%	6	27.3%	22
MA	33	78.6%	9	21.4%	42
ST	29	69.0%	13	31.0%	42
Group 1	222	50.5%	217	49.5%	439
Group 2	113	70.2%	48	29.8%	161
1 & 2	335	55.9%	265	44.1%	600

**Figure 8-14:** Phoncode tested on study 2 errors

**Figure 8-14:** Phoncode tested on study 2 errors

The overall percentage correction for both groups is 55.9%. Group 2 all have higher percentage corrections than Group 1: the Group 2 total is 70.2%, while that for Group 1 is 50.2% ( $p < 0.02$ ). The percentage corrected, for all children, ranges from 35.5% to 78.6%, distributed fairly evenly through the whole range.

### 8.3.2. Errors which the phoncode program failed to correct

Of the 835 misspellings made overall, the phoncode program failed to correct 356 (42.6%). This failure may be attributed to one or more of the following:

1. the misspellings and corrections were not "phonetically equivalent";
2. the program failed to find the "phonetically equivalent" correction for the misspelling, due to:
  - (i) the phoneme-grapheme grammar being incorrect or incomplete;
  - (ii) the segmentation algorithm being incorrect;
  - (iii) the words being incorrectly coded in the phonetically coded dictionary.

In order to determine which of the misspellings might be considered phonetic and which non-phonetic, a judge was used to classify them. This was the same person who was later used to judge the errors that the editcost program failed to correct (see section 8.2.3). The judge was a male Scottish teacher, with a knowledge of linguistics. He was very familiar with the dialect used by the children in the two studies.

The judge was given the complete set of misspellings and corrections, for both sets of children. He was asked to look at each misspelling/correction pair and to decide whether or not they could be considered to be phonetically equivalent: if both were read aloud would they be indistinguishable. After a practice on a set of 'misspellings' and 'corrections' taken from Cohen (1984), the definition was further refined to "both spellings being interpreted as the same word by a local native speaker, when read aloud; the pronunciation of misspellings to be determined by the common pronunciation of graphemes in different contexts". The judge, therefore, was permitted to consider the same misspelling as having more than one pronunciation.

Each error was marked by the judge as either phonetic or non-phonetic. The results of this classification and those of the phoncode program were compared. These results were classified in the following categories:

- (a) correction included in words offered and judged to be 'phonetic' (C/Ph) = agreement;
- (b) correction included in words offered and error judged to be 'non-phonetic' (C/NPh) = disagreement;
- (c) correction not included in words offered and error judged to be 'phonetic' (NC/Ph) = disagreement;
- (d) correction not included in words offered and error judged to be 'non-phonetic' (NC/NPh) = agreement;
- (e) total number of errors

Results of the comparison of the judge's classification and the program performance are given in figures 8-15 and 8-16.

For Study 1, group 1, the agreement between the program and the judge is 78.4% ( $=a + d = 50\% + 28.4\%$ ) and 72.1% for group 2 ( $=32.7\% + 39.4\%$ ): that is, 74.5% ( $=39.2\% + 35.3\%$ ) overall. Groups 1 and 2 showed no significant differences when compared in any of the categories (a),(b),(c),(d). Most disagreement between judge and program outcome occurred in the C/NPh category (22.1%); misspellings classed as non-phonetic by the judge were corrected by the program. Only 1.1% of group 1 errors and 4.8% of group 2 errors (3.4% or 8 errors overall) were classed as phonetic but not corrected. Of the errors made, 39.2% were both classed as phonetic (by the judge) and corrected (by the phoncode program).

For Study 2, 77% agreement is shown between judge and program (group 1 - 76.5%; group 2 - 78.2%). Groups 1 and 2 differed on the frequency of errors classed in categories (a) and (d): group 2 had more errors classed as phonetic and corrected than group 1 ( $p < 0.02$ ) and fewer non-phonetic and non-corrected errors ( $p < 0.05$ ). No significant differences were shown between the two groups in the categories for which judge and program disagreed. Over the two groups 5.8% of errors were classed as phonetic but not corrected (group 1 - 6%; group 2 - 5.6%). In all, 38.7% of errors were classed as phonetic and corrected, with a further 17.2% corrected.

The combined figures for both studies give 76.3% agreement between the program and the judge. 38.8% of errors were judged to be phonetic, and were

	a	b	c	d	e
	C/Ph	C/NPh	NC/Ph	NC/NPh	total
	(% of	(% of	(% of	(% of	number of
	total)	total)	total)	total)	errors
Group 1					
GQ	6	4	0	5	15
	(40%)	(26.7%)	(0%)	(33.3%)	
JM	22	9	0	7	38
	(57.9%)	(23.7%)	(0%)	(18.4%)	
MW	16	5	1	13	35
	(45.7%)	(14.3%)	(2.9%)	(37.1%)	
Group 2					
LB	14	9	0	3	26
	(53.9%)	(34.6%)	(0%)	(11.5%)	
NM	17	4	1	11	33
	(51.6%)	(12.1%)	(3%)	(33.3%)	
CM	15	15	4	26	60
	(25%)	(25%)	(6.7%)	(43.3%)	
SS	2	6	2	18	28
	(7.1%)	(21.5%)	(7.1%)	(64.3%)	
Group 1	44	18	1	25	88
total	(50%)	(20.5%)	(1.1%)	(28.4%)	
Group 2	48	34	7	58	147
total	(32.7%)	(23.1%)	(4.8%)	(39.4%)	
Both	92	52	8	83	235
groups	(39.2%)	(22.1%)	(3.4%)	(35.3%)	

**Figure 8-15:** Comparison of errors corrected by the Phoncode program with those judged to be 'phonetic' - Study 1

corrected, with a further 18.6% corrected (but judged to be non-phonetic). 37.5% were judged to be non-phonetic and were not corrected by the phoncode program. Only 5.1% were judged to be phonetic but not corrected.

The reasons for the failure of the phoncode program will be considered. The program was not designed to correct non-phonetic errors, thus a large percentage of the misspellings (37.5%) were classed as non-phonetic and were not corrected. There were 43 misspellings, judged to be phonetic, which the program failed to correct (NC/Ph). These are listed in figure 8-17.

As stated above, the failure may be attributed to incomplete or incorrect

	a	b	c	d	e
	C/Ph (% of total)	C/NPh (% of total)	NC/Ph (% of total)	NC/NPh (% of total)	total number of errors
Group 1					
FR	52 (42.3%)	26 (21.1%)	12 (9.8%)	33 (26.8%)	123
DV	31 (31.6%)	12 (12.2%)	3 (3.1%)	52 (53.1%)	98
TE	42 (32.1%)	29 (22.1%)	7 (5.3%)	53 (40.5%)	131
DR	20 (23%)	10 (11.5%)	4 (4.6%)	53 (60.9%)	87
Group 2					
GR	25 (45.4%)	10 (18.2%)	5 (9.1%)	15 (27.3%)	55
DI	14 (63.6%)	2 (9.1%)	1 (4.6%)	5 (22.7%)	22
MA	28 (66.7%)	5 (11.9%)	0 (0%)	9 (21.4%)	42
ST	20 (47.6%)	9 (21.4%)	3 (7.2%)	10 (23.8%)	42
Group 1 total	145 (33%)	77 (17.5%)	26 (6%)	191 (43.5%)	439
Group 2 total	87 (54%)	26 (16.2%)	9 (5.6%)	39 (24.2%)	161
Both groups	232 (38.7%)	103 (17.2%)	35 (5.8%)	230 (38.3%)	600

**Figure 8-16:** Comparison of errors corrected by the Phoncode program with those judged to be 'phonetic' - Study 2

grammar; incorrect segmentation; or incorrect coding of the dictionary. The difficulties of segmentation and coding are discussed in chapter 7, subsections 7.3.5, 7.3.6.

Examples of segmentation<sup>1</sup> errors are:

<sup>1</sup>A segmentation error is one where the misspelling is split into graphemes in such a way that it cannot be matched to the phoneme string representing the correction.

Study 1			
MW		NM	
won	one	sounds	souns
SS		CM	
get	cedt	picture	picher
easter	eastr	buttons	butns
		buttons	buttns
		castle	castl
Study 2			
FR		TE	
blood	plood	police	plec
treasure	tresher	seen	cn
diamonds	dimens	dangerously	dangersly
jewels	jouls	ireland	irlnd
using	yoosing	thatcher	thacher
magazine	magzine	work	wrk
magazine	magzeen	if	ifh
computer	compyooter	DV	
how	howe	goals	gois
put	pit	score	score
chemical	cemikle	picture	pichur
plans	plandes	DR	
GR		took	toog
university	univesty	picture	picher
boxes	boxs	kitchen	kitshen
used	yoosed	has	his
alphabet	alphapet	ST	
put	pit	government	goverment
DI		programmes	progames
designed	designed	three	theree

**Figure 8-17:** Errors judged to be phonetic, but not corrected: S1 and S2

y = /y/	u = /ju/	d = /d/	d = /d/
oo = /u/	s = /z/	e = /I/	e = /I/
s = /z/	i = /I/	s = /z/	s = /z/
i = /I/	ng = /ng/	i = /aI/	i = /aI/
ng = /ng/		g = ?	gn = /n/
		nn = /n/	ed = /d/
		ed = /d/	
k = /k/	k = /k/	s = /s/	s = /s/
i = /I/	i = /I/	c = /k/	c = /k/
t = /t/	tch = /ch/	o = /o:/	o_e = /o:/
sh = /sh/	e = /E/	rr = /r/	r = /r/
e = /E/	n = /n/	e = ?	
n = /n/			

The phoneme-grapheme grammar failed to provide matches in a number of cases, though for some of them their classification as 'phonetic' errors might be disputed.

Examples of these are:

get	cedt	blood	plood
put	pit	took	toog
has	his	alphabet	alphapet

Other classes of errors that presented difficulties include:

- (a) omitted schwa, particularly before n and l  
e.g. buttons buttns police plec
- (b) other omitted vowels  
e.g. boxes boxs chemical cemikle
- (c) errors involving 'r'  
e.g. easter eastr picture picher
- (d) consonant confusions, particularly involving 'd', 't', 'ch'  
e.g. get cedt picture picher
- (e) consonant omissions, particularly d after n  
e.g. sounds sounds diamonds dimens

The other set of misspellings that judge and program disagreed on were those judged as non-phonetic, but corrected by the phoncode program. A large number of these were vowel confusions accepted as equivalent by the phoncode grammar but rejected by the judge. Additionally, other classes of errors accepted by the program, but considered 'non-phonetic', were:

- \* errors involving 'r' (and vowel);
- \* final 'e' (omitted and added);
- \* transpositions, in particular 'ed/de' and 'es/se' and vowels;
- \* incorrectly doubled or singled consonants, in particular 'n' before 'g' or 't', and 'l' before 'k' or 'd';
- \* errors involving 'h' (usually silent)

For some of these misspellings, the alteration of a grapheme from 'tied' to 'untied' would enable them to be corrected, and matched to their phonetic

equivalent. For a number of others, in particular those involving omission of an unstressed vowel, the program would need to be altered to take them into account.

Summarising the results for the phoncode program overall:

- \* the program succeeded in correcting

- (a) 61.3% of errors made by Group 1, when tested;

- (b) 55.9% of errors made by Group 2, when tested;

- (c) 57.4% of errors tested overall.

- \* of those it failed to correct (356 errors)

- (i) 43 were judged to be phonetic (therefore attributable to failure on the part of the program), accounting for 5.1% of errors overall;

- (ii) 313 were judged to be non-phonetic, 37.5%;

additionally, 38.8% of misspellings were both judged to be phonetic and corrected by the phoncode program.

### 8.4. Results for combined programs

The results of testing the performance of the two programs, on the sets of misspellings from the two studies, were combined. There was a large amount of overlap between the corrections. The results for each program and for the combined programs are given in figures 8-18 and 8-19.

	a	b	c	d	e
	corrected by editcost	corrected by phoncode	corrected by neither	corrected by combined	total number of errors
Group 1					
GQ	15 (100%)	10 (66.7%)	0 (0%)	15 (100%)	15
JM	38 (100%)	31 (81.6%)	0 (0%)	38 (100%)	38
MW	29 (82.9%)	21 (60%)	5 (14.3%)	30 (85.7%)	35
Group 2					
LB	24 (92.3%)	23 (88.5%)	0 (0%)	26 (100%)	26
NM	30 (90.9%)	21 (63.7%)	2 (6.1%)	31 (93.9%)	33
CM	46 (76.7%)	30 (50%)	10 (16.7%)	50 (83.3%)	60
SS	18 (63.4%)	8 (28.6%)	9 (32.1%)	19 (67.9%)	28
Group 1 total	82 (93.2%)	62 (70.5%)	5 (5.7%)	83 (94.3%)	88
Group 2 total	118 (80.3%)	82 (55.8%)	21 (14.3%)	126 (85.7%)	147
Both groups	200 (85.1%)	144 (61.3%)	26 (11.1%)	209 (88.9%)	235

**Figure 8-18:** Comparison of errors corrected by Editcost and by Phoncode programs - Study 1

For each child, for each group, the following information is given:

- (a) the number and percentage of errors corrected by the editcost program;
- (b) the number and percentage of errors corrected by the phoncode program;

- (c) the number and percentage of errors corrected by neither program;
- (d) the number and percentage of errors corrected by either of the two programs;
- (e) the total number of errors made.

	a	b	c	d	e
	corrected by editcost	corrected by phoncode	corrected by neither	corrected by combined	total number of errors
Group 1					
FR	103 (83.7%)	78 (63.4%)	10 (8.1%)	113 (91.9%)	123
DV	65 (66.3%)	43 (43.8%)	22 (22.4%)	76 (77.6%)	98
TE	106 (80.9%)	71 (53.2%)	13 (9.9%)	118 (90.1%)	131
DR	55 (63.2%)	30 (34.5%)	26 (29.9%)	61 (70.1%)	87
Group 2					
GR	45 (81.2%)	35 (63.6%)	6 (10.9%)	49 (89.1%)	55
DI	21 (95.5%)	16 (72.7%)	1 (4.5%)	21 (95.5%)	22
MA	39 (92.9%)	33 (78.6%)	1 (2.4%)	41 (97.6%)	42
ST	39 (92.9%)	29 (69%)	2 (4.8%)	40 (95.2%)	42
Group 1 total	329 (74.9%)	222 (50.5%)	71 (16.2%)	368 (83.8%)	439
Group 2 total	144 (89.4%)	113 (70.2%)	10 (6.2%)	151 (93.8%)	161
Both groups	473 (78.8%)	335 (55.9%)	81 (13.5%)	519 (86.5%)	600

**Figure 8-19:** Comparison of errors corrected by Editcost and by Phoncode programs - Study 2

For Study 1, the percentage correction for the combined programs is 88.9%. Of the 35 errors that the editcost program failed to correct, 9 errors were corrected by the phoncode program. The remaining 26 that neither program corrected include some that were neither corrected by the judge (in testing editcost) nor judged to be phonetic. Group 1 show a higher percentage

correction in all categories than group 2, though none of the differences are significant. By combining the two programs the number of errors corrected is increased, for most children. GQ and JM are the exceptions with 100% correction using the editcost program alone. The increases vary from one additional correction (MW, NM, SS), to two (LB), to four (CM).

For Study 2, the combined programs correct 86.5% of misspellings. 46 are corrected by the phoncode program that were not corrected by the editcost program, leaving 81 misspellings not corrected by either program. Group 2 show higher percentage corrections than group 1 for the individual programs ( $p < 0.05$  for editcost and  $p < 0.02$  for phoncode) but no significant differences for the combined programs. Improvements in the number of misspellings corrected vary from 0 (DI), 1 (ST) to 11 (DV), 12 (TE).

The overall percentage correction by the combined programs is 87.2%.

### 8.5. Results for individual children

In this section the performance of the spelling correction program, in relation to individual children is considered. The relationships between a number of measures was found by correlation of the rankings of individual children on performance measures. It was hypothesized that the children who made the most 'regular' errors, i.e. those who produced the fewest bizarre spellings, would also be those for whom the editcost and phoncode correctors would be most successful. Additionally, the errors that they make would be considered to be 'phonetic'. The children making the most 'regular' errors were those who were perceived as having the least difficulty.

The children were ranked (roughly and subjectively, it should be noted) in terms of their spelling ability. This ranking was based on observation by the investigator and discussion with the Reading Unit teacher. For S1, the rough rankings in order of decreasing ability, were:

GQ; JM; LB; MW; NM; CM; SS

For S2 the rough rankings were:

MA and DI; ST; GR; FR; DV; DR; TE

The hypotheses tested were:

1. success of correction by the editcost and the phoncode programs would correlate;
2. children whose errors were judged to be phonetic would also show greatest success with the phoncode program;
3. the children ranked as most able would be those for whom the programs were most successful and whose errors were judged to be phonetic.

For the children in each group, the relationships between the following measures were found using the Spearman Rank correlation coefficient.

- (a) percentage correction by the editcost program (in testing);
- (b) percentage of corrections that were off(1);
- (c) percentage correction by the phoncode program;
- (d) percentage of errors judged to be phonetic;

- (e) percentage improvement of editcost results when both programs' results are combined.

The perceived rankings of the children's general spelling ability were not statistically correlated with these measures as they were considered to be too subjective and crude. They are, however, considered in relation to the results of these correlations. The measure of b) was included to test whether there was any relationship between the degree of success of the editcost program (where off(1) indicated greatest success) and other measures. Measure e) was included to further test the relationship between the editcost and phoncode programs' results.

For all measures, percentages were of total number of errors made by each child (except (b), which was percentage of (a)).

Significant correlations were found between a number of measures. These will be summarised and then discussed.

#### For Study 1

- correlation between	a) and c) = .88	( p < 0.05 )
	b) and d) = .76	( p < 0.05 )
	c) and d) = .75	( p < 0.05 )

#### For Study 2

- correlation between	a) and c) = .93	( p < 0.01 )
	c) and d) = .98	( p < 0.01 )
	a) and d) = .97	( p < 0.01 )
- correlation between	b) and e) = -.76	( p < 0.05 )
	b) and a) = .82	( p < 0.05 )
	b) and c) = .68	( p < 0.05 )
	b) and d) = .71	( p < 0.05 )
	e) and a) = -.72	( p < 0.05 )
	e) and c) = -.81	( p < 0.05 )
	e) and d) = -.74	( p < 0.05 )

For children in Study 1, success of editcost and phoncode programs were correlated; as were success of phoncode program and percentage of errors judged to be phonetic, and percentage corrections offered as the first editcost option and percentage judged to be phonetic.

Stronger correlations are shown for Study 2: performance of phoncode and editcost programs, and percentage of errors judged phonetic all correlate. Additionally, percentage of errors offered as first option correlated negatively with the percentage improvement made by the phoncode program when both program' results were combined: both of these correlate (the latter, negatively) with the three strongly correlated measures above.

Therefore, in general it can be said that those children for whom the editcost program is successful, the phoncode program will also be successful. A large part of the failure of the editcost program can be attributed to unrecognizable errors. These children also make the fewest unrecognizable errors. The correlation between phoncode performance and judgement of phonetic errors suggests that those children for whom the phoncode program is most successful make the fewest non-phonetic errors. These relations are shown most strongly in the Study 2 children; a strong direct correlation is also shown between performance of editcost program and percentage of phonetic errors. For these children, the correlations also suggest that those with the most errors offered as first options also make most phonetic errors, and fewest non-phonetic errors.

The negative correlation between measures e) and a) is to be expected: the more successful the editcost program is, the less scope there will be for improvement. The editcost program incorporates some information relating to phonetic equivalence of words (e.g. most likely substitutions are phonetically similar), hence the high correlation between measures a) and c) is also not surprising.

Considering the individual performance rankings of the children, firstly for Study 1; group 1 were described as the more able students (see notes on children in appendix A), and group 2 as the "hopeless cases" (with LB as an addition to this group). From the performance rankings, GQ, JM and LB generally come out as the top group, with MW and NM as the middle group (except for a percentage of errors judged phonetic, where NM and GQ swop groups), and CM and SS as the least able, with the worst results for all measures. These rankings fit very well with the perceived abilities of the children. For Study 2, the performance rankings are even clearer: best ranked are DI, MA and ST, then FR and GR

(where group 1 - "moderately able", and group 2 - "very bright", overlap), and finally TE, DV, and DR. Again, there is a good fit between rankings and perceived abilities, with the exception of TE who performs better than would be expected.

In relation to the theoretical discussion of the stages of failure in the spelling process, various inferences may be made on the basis of these findings. Children who have the least difficulties are more likely to be failing at a later stage in the process than those who make a large number of bizarre and irregular spelling errors. If the former succeed at the 'selection of plausible graphemes' stage, but fail at the third stage, their errors will be phonetic. They are more likely to be using correspondences from the phoncode grammar and hence their errors will be corrected by the phoncode program. Their errors are occurring in the selection of orthographically correct plausible graphemes: information relating to the format is used by the editcost program to correct these successfully. It is expected that both editcost and phoncode programs will successfully correct the errors made by these children. In terms of absolute success, it can be seen from the results that the editcost program is clearly more successful. It is designed to cope with both phonetic and non-phonetic errors: hence its higher rate of success.

Where there is failure at the first or second stage, that is, the graphemes selected to represent the speech sounds are not plausible, we would expect the phoncode program to fail. We would also expect a lower rate of phonetic errors. The editcost program is able to 'pick up' some of these non-phonetic errors: some are too irregular, however, and cannot be fitted into any general description of errors.

Those children perceived as 'better spellers' showed more regularity in their errors, made fewer non-phonetic errors and were more likely to have their errors corrected successfully by both the editcost and phoncode programs. They were considered to be failing to select the correct grapheme from the plausible graphemes generated.

The children perceived as least able showed more irregular errors and more non-phonetic errors. The editcost program was more successful for them than

the phoncode program. Neither were as successful with these children as with the better spellers. Their failings occur at the first or second stage in the spelling process; that is, in the segmenting of the word into phonemes, or in the selection of plausible graphemes to represent each phoneme. Inferences cannot be drawn from these results to judge at which of the first two stages the failing is occurring.

It might be inferred from these findings that success in correction by the phoncode program implies that a phonological strategy is being used by the child. Conversely, success in correction by the editcost program could be taken to suggest that a visuo-orthographic strategy is being employed. If this argument is accepted, the implication would be that those children for whom both programs are successful used both phonological and visuo-orthographic strategies in spelling. Following from this, it could be argued that the children for whom the phoncode program is comparatively less successful use predominantly visuo-orthographic strategies. There are no clear conclusions that can be drawn from the evidence presented here, however, for two reasons:

1. the editcost program incorporates a certain amount of phonological information in relation to likely errors: therefore, the success of the editcost program and the failure of the phoncode program does not necessarily imply that a phonological strategy is not being used;
2. it is very difficult to assess "comparatively less successful": whilst the rankings on editcost and phoncode performance correlate highly, the absolute differences between percentages appear to bear little relation to these rankings.

One conclusion that may be drawn is that the more able children appear to use both strategies with more success than the less able children.

### 8.6. Testing the programs on independent data

The editcost and phoncode programs were also tested on data from an external source. These were a corpus of misspellings of thirty words produced by 202 ten-year old children, in a dictation test. The children were a random sample selected from a group of 15,000 children in English and Welsh schools. The data was made available to Roger Mitton (Birkbeck College, London) by Dr. Uta Frith (MRC Cognitive Development Unit, London). A copy of the corpus of misspellings was provided for testing in this thesis.

The number of misspellings in the corpus is 2482. Of these 1364 are unique: the rest are the same misspelling made by more than one child. The set of unique misspellings will be referred to here as 'errors excluding repeats', whilst the full corpus will be referred to as 'errors including repeats'. The set of errors excluding repeats was used with the editcost and phoncode programs. The testing dictionary was that referred to elsewhere in this thesis (section 8.2.2), with the addition of those of the thirty dictated words that were not already included.

Results are given for each of the thirty words: figure 8-20 shows the results for the errors, excluding repeats; figure 8-21 shows correction of errors including repeats.

Results of testing are given in the following categories:

- (a) the number of errors for which the correction was offered by the editcost program (and the number for which these were the first offer, when offered);
- (b) the number of errors for which the correction was offered by the phoncode program;
- (c) the number of errors for which the correction was offered by either of the two programs;
- (d) the percentage of the total number of errors for which the correction was offered, by either program (c/e);
- (e) the total number of errors.

For 78.2% of the unique errors, and for 86.3% of the total number of errors, the

Words	a corrected by editcost (off(1))	b corrected by phoncode	c corrected by either (number)	d corrected by either (%)	e total number of errors
often	26 (19)	15	26	89.7%	29
visited	41 (33)	8	41	91.1%	45
aunt	14 (9)	4	14	66.7%	21
magnificent	83 (78)	21	83	82.2%	101
house	8 (5)	3	8	88.9%	9
opposite	56 (44)	30	58	79.4%	73
gallery	51 (26)	18	51	81%	63
remember	37 (31)	9	37	90.2%	41
splendid	33 (29)	9	33	58.9%	56
purple	24 (18)	12	25	75.8%	33
curtains	39 (32)	24	39	79.6%	49
wrote	13 (8)	7	14	56%	25
poetry	62 (46)	24	63	78.8%	80
problem	35 (30)	8	35	83.3%	42
understand	24 (20)	5	24	82.8%	29
latest	32 (27)	10	32	71.1%	45
poems	28 (23)	9	29	74.4%	39
wanted	10 (5)	4	11	52.4%	21
laugh	18 (9)	9	23	62.2%	37
pretend	45 (37)	10	45	81.8%	55
really	29 (19)	15	29	70.7%	41
special	74 (53)	32	74	85.1%	87
refreshment	53 (48)	16	53	81.5%	65
there	5 (2)	4	5	71.4%	7
blue	5 (4)	3	5	71.4%	7
juice	18 (14)	23	32	69.6%	46
cake	9 (6)	2	11	84.6%	13
biscuits	63 (55)	16	63	80.8%	78
stomach	62 (44)	44	67	79.8%	84
contented	37 (33)	9	37	86%	43
Total	1034 (807)	403	1067	78.2%	1364

**Figure 8-20:** Testing of the editcost and phoncode programs on independent data - excluding repeats

correction is offered by either the editcost or the phoncode program. Of the errors corrected by the editcost program, 85.7% are offered as the first option i.e. the least cost repair, representing 71.7% of the total number of errors.

As with the children in the two studies, the editcost program was more

Words	a	b	c	d	e
	corrected	corrected	corrected	corrected	total
	by editcost (off(1))	by phoncode	by either (number)	by either (%)	number of errors
often	51 (42)	32	51	91.1%	56
visited	93 (78)	24	93	93.9%	99
aunt	71 (63)	43	71	87.7%	81
magnificent	136 (131)	59	136	88.3%	154
house	14 (11)	9	14	93.3%	15
opposite	125 (109)	97	132	88%	150
gallery	101 (70)	58	101	88.6%	114
remember	90 (84)	13	90	94.7%	95
splendid	102 (96)	61	102	81.6%	125
purple	41 (35)	27	42	84%	50
curtains	76 (66)	57	76	88.4%	86
wrote	63 (55)	60	64	78%	82
poetry	91 (73)	37	92	84.4%	109
problem	66 (60)	15	66	90.4%	73
understand	29 (25)	5	29	85.3%	34
latest	47 (42)	14	47	77%	61
poems	64 (58)	19	65	86.7%	75
wanted	28 (17)	16	29	70.8%	41
laugh	28 (18)	30	50	76.9%	65
pretend	81 (69)	26	81	88%	92
really	110 (99)	94	110	90.2%	122
special	108 (87)	52	108	89.3%	121
refreshment	77 (72)	28	77	86.5%	89
there	19 (12)	19	19	54.3%	35
blue	11 (10)	3	11	84.6%	13
juice	54 (50)	64	74	83.1%	89
cake	13 (10)	5	18	90%	20
biscuits	126 (113)	65	126	89.4%	141
stomach	108 (77)	86	116	84.7%	137
contented	52 (47)	16	52	89.7%	58
Total	2075 (1779)	1134	2142	85.7%	2482

**Figure 8-21:** Testing of the editcost and phoncode programs on independent data - including repeats

successful than the phoncode program. Because a large number of the errors made by the children would probably not be classed as phonetic, this was to be expected. Some failure could be attributed to the program, however. The words that the correctors failed on are not analysed in detail, though the discussion of failure in relation to the two studies is of relevance (see section

8.3.2). The phoncode program provided little improvement over the editcost program, except for the words 'laugh' and 'juice'.

The combined programs failed to achieve 70% correction on unique misspellings of 'aunt', 'wrote', 'wanted', 'laugh', 'juice' and 'splendid'. There is improvement in performance when repeated errors are included. Those errors that the program succeeds in correcting are those that are most often repeated (the exception being those of 'there'). For 7 of the 30 words, more than 90% of the misspellings were corrected.

Mitton had previously tested two other spelling correction algorithms with this data (Mitton, 1984b). He found that 42% of errors (including repeats) would be included as candidates when classed as single edit misspellings (i.e. one edit operation required to correct the error). Depending upon the size and the content of the dictionary, there may be many other candidates. The errors were also coded using the soundex code. For 64% of errors the coding matched for error and correction. Again, many other candidates may also match. Combining the results of the two algorithms, the correction was found to be in the candidate list for 72.9% of errors. For the editcost program alone, the percentage of errors corrected was 83.6%, of which for more than 85% the first word offered was the correction (=71.7% of total). The correction programs, therefore, though designed for use with children with spelling difficulties, could also be used by other children.

## 8.7. Summary

The results presented in this chapter show that the spelling correction programs, developed in this study, were successful in correcting the errors made by children with learning difficulties in spelling. The editcost program was the more successful of the two. As it was designed to deal with both phonetic and non-phonetic errors, whereas the phoncode program was designed to deal with phonetic errors, this was to be expected. The editcost program succeeded in offering corrections for more than 80% of the errors made in the two studies. The phoncode program succeeded in offering corrections for 57.4% of errors tested. Of those the phoncode program failed to correct (42.6%), 37.5% were judged not to be phonetic. In combination the two programs provided corrections for 87.2% of errors over both studies. The success of the programs is restricted by the correction being in the dictionary: if it is not in the dictionary it cannot be offered to the user.

The programs were also tested on independent data and found to be successful: 78.2% of unique errors made were corrected by the combined programs; 86.3% of errors in the corpus (including repeats) were corrected. Of the corrections made by the editcost program, for 71.7% (of the complete corpus) the intended correction was the first word offered. This compares favourably with other algorithms tested on the same data. The program would, therefore, be suitable for use by children with no specific difficulties.

Evidence is provided that there are regularities in the errors made by children with spelling disabilities. In testing the editcost program, 13.7% of failures to correct errors were attributed to there being insufficient regularities to enable correction. 80.6% of errors were successfully corrected and 5.7% of failures were attributed to failure on the part of the program. Thus, for 86.3% of errors there was sufficient regularity in the misspelling to permit correction of the error.

In considering the results of the phoncode program, 57.4% were corrected overall and a further 5.1% attributed to failure on the part of the program. 37.5% of errors, therefore, were assessed as being non-phonetic – that is, phoneme-grapheme correspondences on which they were based did not

conform to the grammar provided. However, 62.5% of misspellings did conform to the grammar. It is argued that there are regular phoneme-grapheme correspondences in the children's spellings, and that there are also additional regularities in the orthography (according to the additional corrections by the editcost program).

That the programs succeed in correcting a large proportion of the errors made also demonstrates that these regularities can be used by the programs to reconstruct the corrections from the errors. The information incorporated in the programs, based on the description of the errors in terms of format, general classes of characters and rules, and phoneme-grapheme correspondences, enables successful debugging of the error to provide the correction. The description of errors in these terms is also, to a large extent, validated by the results.

## Chapter 9

### Conclusions

#### 9.1. Introduction

This thesis set out to address a number of questions, both theoretical and practical. Answers to these questions are provided in this chapter. General and specific criticisms of the research are also presented and further work is proposed.

#### 9.2. Theoretical Questions

The following questions, relating to the children's difficulties with spelling, were addressed.

- \* Do the errors made by the children show regularity, or are they random?
- \* If a classification scheme is developed, based on the errors, can it provide adequate information to enable a computer program to 'debug' the children's errors and to reconstruct the intended words?
- \* Are the children able to recognize intended words (correctly spelt) even if they cannot produce them?

In the review of the literature (chapter 2) it was argued that there were regularities in English Spelling, both at the level of phoneme-grapheme correspondences and at other levels. In chapter 8 this question was addressed directly, in relation to children with learning difficulties in spelling. It was found that for 86.3% of the errors made by children in the two pilot studies there was sufficient regularity to permit the misspellings to be corrected. Nearly two-thirds of the errors were corrected by the phoncode program, providing evidence that there are regular phoneme-grapheme correspondences in the children's spelling. The additional errors corrected by the editcost program

demonstrate that there are other regularities, relating to other features in the orthography. It was concluded that there was evidence to show regularities in the children's spelling.

It was further concluded in chapter 8 that adequate information was provided to enable the intended word to be reconstructed from the misspelling. The classification scheme (described in chapter 4) provided a description of the errors in terms of format, general classes of characters and rules, and phoneme-grapheme correspondences. This description was used to analyse a number of errors made by the children in pilot study 1. The information obtained was incorporated in the spelling correction programs. This enabled the programs to 'debug' a large number of the children's errors and to provide the reconstructed correction.

The observation was made, in chapter 6, that the children were able to recognize the intended word even when they could not produce it. The editcost correction program was used by children from the Reading Unit (PS2). In 87.7% of cases where the correct word was offered by the program it was selected by the child as the intended word. In cases where the word was not offered, in only 11% did the child not realise this. This suggests that the child has access to some representation of the word that is sufficient to permit recognition of the word but not to enable production.

### **9.3. Practical Questions**

Questions relating to the practicality of a computer program as a tool were also considered.

- \* If a computer-based tool is provided to help the child:
  - is he able to use it?
  - is he willing to use it?
- \* How well does the tool succeed in correcting his errors?
- \* Do the answers to these two questions vary according to the individual using the tool?

The questions of whether the child was willing and able to use the computer program as a tool were addressed in chapter 6. The following conclusions were drawn:

- \* that children with learning difficulties in spelling were both willing and able to use a text editor to write stories;
- \* that when a spelling correction program was provided it was used by the children to check the spelling of the words they were uncertain of.

The design of a program to incorporate both editcost and phoncode correctors and a text editor was presented in chapter 5. This program has not yet been implemented, however.

The performance of the editcost and phoncode programs, in correcting errors, was discussed in chapter 8. Both were successful in correcting the children's errors, the editcost program being the more successful of the two. Summarizing the results for the editcost program:

- \* the program succeeded in correcting
  - (a) 85.1% of errors made by Pilot Group 1, when tested;
  - (b) 78.8% of errors made by Pilot Group 2, when tested;
  - (c) 91.1% of errors made by Pilot Group 2 (for which the correction was available) when the program was in use;
  - (d) 80.6% of errors tested (a + b) overall.
- \* of those it failed to correct (162 errors)
  - (i) 48 were corrected by the judge (therefore attributable to failure on the part of the program), accounting for 5.7% of errors overall;
  - (ii) 114 were not corrected by the judge (therefore attributable to insufficient regularities shown in the errors), accounting for 13.7% overall.

Individual children did adopt different strategies when using the computer tools. These were noted in chapter 6. In general the tools were used as designed, although the lookup facility was used in preference to the editcost checking

facility by one child. The performance results for individuals varied. These were discussed in chapter 8. The conclusions drawn from these (in relation to the less able spellers) were that the children perceived as the more competent spellers made fewer non-phonetic errors, and had a higher percentage of errors corrected by both the phoncode and editcost programs. It was suggested that the failings of the more able children were at the stage of the selection of the appropriate grapheme from a number of plausible ones. The less able children, who showed more irregular errors and more non-phonetic errors, could be failing at either the first or second stage in the spelling process – that is, at the stage of converting speech into phonemes, or at the selection of plausible graphemes to represent the phonemes. It was also suggested that the more able children might be using phonological and visuo-orthographic strategies in spelling, whilst the less able children were using only one of these, or using both less effectively.

#### **9.4. Additional Questions**

Additional questions, not addressed directly, but also considered in this thesis:

\* Could such a tool provide us with:

- information about the phoneme-grapheme correspondences used by individual children?
- other information that might be of use to the remedial teacher?

\* Could the tool be used by other groups of children?

##### **9.4.1. Information about the child**

At present, the correction programs use the information relating to the error classification but this information is not made explicit. It would be possible to extend both programs to provide more information about the errors made, in addition to correcting them.

In order to model the child, information about both correct spelling and errors is needed. A model might contain:

\* general phoneme-grapheme correspondence rules,

- \* rules relating to morphology, syntax, permitted letter sequences, known irregularities, etc.

- \* information about the child's errors, relating to 'bugs' in these rules.

The work by Hanna et al (Hanna et al, 1966) is an attempt to provide a model of correct spelling, though it requires further information to be a complete and correct model. Ways in which the errors might be modelled are considered here.

Information about errors could be used to describe an individual child's difficulties. The sort of information that might be provided about the child is as follows:

- \* frequent b/d confusions
- \* omission of final modifying e
- \* omission of medial vowels representing schwa
- \* transposition of vowels modified by l
- \* replacement of d by t when preceded by vowel and followed by schwa
- \* confusion involving vowels modified by r

Information to be used in providing a model of the child's errors could be obtained by use of the editcost and phoncode programs. The former could provide a description of the errors in terms of edit operations and general classes of characters involved, providing classification in terms of general classes of errors. These could be related to bugs in the rules of visuo-orthographic structure. The phoncode program could be used to describe errors in terms of phoneme-grapheme correspondences for each child: in cases where inappropriate graphemes are used for phonemes, either:

(a) incorrect choice of grapheme for the phoneme, or

(b) incorrect choice of phoneme,

can be inferred.

For both programs, there will be different ways of describing the same errors.

Some selection will be required of the most appropriate way to describe each error, (i.e. the most frequently occurring buggy rules or correspondences). This problem is similar to that experienced in the Buggy research, where decisions were made on how to select the minimal set of rules to best describe the errors made by a particular child.

The description of each child's errors can provide an indication of the stage at which the child fails: he may be using inappropriate graphemes to represent certain phonemes, or he may be using appropriate phonemes in the wrong contexts. More detailed information about the specific contexts would also be available.

How each of the programs would be used, to provide information about each child's bugs, will be considered briefly.

The set of error-correction pairs, for each child, would be recorded. Each pair would be compared using the editcost program. The cheapest path through the editcost graph would be reconstructed, and the sequence of edits made noted. Information about the immediate context of each edit would also be saved. The sequence of edits and context information for all error correction pairs would be collected. From these, the minimal set of rules describing bugs in the child's spelling could be inferred. The full set of possible buggy rules, or inappropriately applied rules, would need to be generated in order that those for each individual could be selected. Identification of this set, based on work done in this thesis, is proposed as a topic for further research.

Each pair would also be analysed in terms of phoneme-grapheme correspondences used. A check would first be made to see if, according to the phoncode program, the error could be considered to be phonetically equivalent to the correction. This would provide graphemic correspondences for the correct phonemes in the word. These would be 'appropriate' phoneme-grapheme correspondences. In cases where there is not a match made by the phoncode program, a 'best match' would be attempted. Decisions would have to be made about those parts of the word where the graphemes are appropriate for the phonemes represented, and those where errors have been made. This would be done in combination with the editcost program. In places

where letters are thought to be omitted (according to the sequence of edits produced by editcost) it might be inferred that a phoneme has been omitted completely from a word. This would be decided by the closeness of the match provided by phoncode: it might also be the case that no phonemes are omitted, but that they are misrepresented.

Taking the full set of error-correction pairs, and the phoneme-grapheme correspondences produced by these, a grammar could be constructed specific to the individual child. This would consist of both a subset of the phoncode grammar plus the erroneous phoneme-grapheme correspondences found to be commonly used by the child.

In general, when constructing a model of the child's spelling, input will be required from both the editcost and phoncode programs. They are currently constructed in such a way that this would be feasible, but need to be extended further to permit it. Ideally, they would need to interact with each other, which at present they do not do.

#### 9.4.2. Providing information for the teacher

If the programs were extended in the ways described above, information about each child's errors could be provided for the teacher. This information could be used in the analysis of each child's specific difficulties, and to provide an indication of where remedial teaching might be focussed.

A summary of the errors made by each child might, on its own, be useful. If the more general spelling checking program described in chapter 5 were implemented it could be extended to detect errors as well as offering the facility to correct them. The difficulties of error detection were discussed in chapter 3. Again, this might be of use to the teacher: it might provide automatic checking of the child's work (though automatic correction would be less desirable).

If a set of 'correct' spelling rules were available, it might be possible for the teacher to ascertain (with the help of the programs) what the child does know about spelling – that is, those regularities and correspondences that each uses correctly. This could form part of a profile of the child. This profile would be

produced as a direct result of the child's interaction with some extended version of the two correction programs combined. It would include buggy rules and correspondences as well as correct ones. A specific profile would be produced for each child, for the teacher's reference.

It is conceivable that the tool might also be extended to provide on-line remediation (as the next stage on from error detection, error correction, and error classification). Existing remedial programs were reviewed in chapter 2: it is not clear that any would be suitable for adaption. However, as a tool for detection of errors, on-line help with correction and the provision of a description of the child's errors, such tools could form a useful aid to teacher and child.

#### 9.4.3. Use by other groups

The correction programs were tested on errors made by children who had no specific spelling difficulties (described in chapter 8). They were successful in correcting a large proportion of the errors. The programs might, therefore, be of use to groups of children other than those it was designed to help. Their use with other groups could be tested. Two groups that might find them of special interest are those adults that have spelling difficulties, and younger children. Additionally, extended versions of the programs might also be found to be of interest to children with fewer spelling difficulties. Other tools, such as text editors and story planning aids have been used previously: interactive spelling correctors could be a useful addition.

### 9.5. Criticisms

There are a number of criticisms that can be made of the work presented in this thesis. Some of them will be discussed in this section.

The performance of the editcost and phoncode programs is far from perfect. In particular, the phoncode program fails to correct a substantial proportion of the errors made. This is to be expected: the phoncode program is not designed to correct non-phonetic errors, though it does succeed in correcting a substantial number of errors that were considered, by the judge, to be non-phonetic. Whilst it is used in conjunction with the editcost program (in testing), the

success rate is improved. However, the editcost program used alone would be almost as successful. If the two programs could interact with each other a more successful and efficient program might result. How this might be done is not considered here. Each program incorporates different information, using it in different ways to correct spelling. There is some overlap between the two: the editcost program incorporates a certain amount of phonemic information that helps it in the selection of corrections.

An important use of the phoncode program is in the classification of errors as phonetic/non-phonetic, and for providing information about phoneme-grapheme correspondences, as opposed to its use in spelling correction. It provides information that cannot be produced by the editcost program alone.

A further problem found, when the children in PS2 used the editcost correction program, was the time it took to find the correction. This time increases with the number of words on the shortlist, to be compared with the misspelling. More efficient program code might be written to improve the program performance or implementation in another language might make it run faster. Better methods of shortlisting the words to be compared and provision of a maximum for the cost of editing might improve the editcost performance.

Comparisons have not been made of the speed of the editcost and phoncode programs. Should the phoncode program turn out to be much faster, then the case could be argued for checking the spelling with the phoncode program initially, and only using the editcost program to check and correct when this fails. The dictionary size could also be reduced to increase speed: a larger stored dictionary would be checked after checking in the small local dictionary had failed. If this involved more decisions from the child, it might not be successful.

Alternatively, Yannakoudakis and Fawthrop's program, described in chapter 4, might be adapted for use. It is not clear, without further investigation, how suitable this would be. It is also possible that, as their program is a commercial product, this would not be permitted.

As stated above, the programs do not currently provide explicit information

relating to the classification of the errors. The possibility of extending the programs to do so has been discussed.

It is acknowledged that there is not a clean and direct correspondence between Frith's three stages in the spelling process, the levels of classification proposed, and the information used by the editcost and phoncode programs. However, it is believed by the author (and argued in chapter 4) that there is sufficient correspondence between the three to be of interest. The relationships should be investigated by further work, particularly in respect of those children failing at the first and second stages in the spelling process.

Whilst distinctions have been made in the literature between Poor Spellers/Poor Readers and Poor Spellers/Good Readers this distinction was not maintained in this thesis. This was mainly due to the decision to focus on the classification of spelling errors and on tools as aids for spelling disabilities. It might have been of benefit to have considered the scores obtained by the children on standardized spelling tests and reading tests, and to have incorporated this information in the analysis. Further information in relation to the children's attitudes towards taking part in the study, and to any effects of taking part in the study that might have carried over into the classroom, might have been of interest.

The form that is taken by the graphemic code, used in the representation of words, is an issue that is discussed in the psychological literature. No opinion is ventured on it here. It is an issue that needs addressing in a rigorous experimental study, and it was felt that it was not appropriate to address it in this thesis. In the same vein, it was not felt to be appropriate to address the problems of transfer from typing to written spelling.

No evidence was presented in this study to show improvement in the children's spelling ability. The major concerns here were collecting data on the children's errors; incorporating it in the correction programs; seeing how these programs might be used and, in doing so, obtaining further information about the children's errors. Further work would be required to carry out a full scale evaluation of the programs as tools, in terms of improvement in performance. In particular, factors such as the increase in motivation due to the novelty of coming to the D.A.I. and using the computer would have to be controlled for.

More specific methodological criticisms can be made. The use of one person to judge the spelling errors can be criticised. Time constraints prevented the use of more judges, though a wider consensus would have been achieved on what constitutes a 'phonetic' error and an 'unrecognizable' error. The main concern in this study was to obtain a judgement independent of that of the author: it is believed that this was achieved.

## **9.6. Proposals for further research**

Extensions of the programs to enable classification of errors have already been outlined above. Others ways in which they might be used by a teacher have also been suggested and use by other groups of children has been proposed.

In response to the criticisms made in the preceding section, the programs might be evaluated more formally, in use in the classroom.

The programs currently reconstruct the intended word from the error. If the classification information were available, the programs could be further extended to generate errors from correct words. This facility might be used to simulate the spelling performance of the individual child and to test general theories in relation to the performance.

A further extension of the editcost program would be to 'tune' the weightings of the edit operations to individual children. If errors were analysed at the format level, the weightings could be adjusted to reflect those errors made most frequently by the individual child: the program could be 'tuned' to any one child. The program is currently implemented in such a way as to facilitate this.

## **9.7. Summary of conclusions**

The following conclusions are drawn in this thesis:

- (a) The errors made by children with learning difficulties in spelling show regularities in both the phoneme-grapheme correspondences and at the level of the orthography.
- (b) The classification scheme developed, based on the children's errors, provides a description of these errors at three levels: the format level; in terms of general classes of characters and rules; and in terms of phoneme-grapheme correspondences.

- (c) It provides adequate information to enable a computer program to 'debug' the children's errors and to reconstruct the intended words.
- (d) Computer tools in the form of interactive spelling correctors have been developed, and used by the children when writing.
- (e) These tools are able to offer a correction for a substantial proportion of the child's errors. They are more successful with more able children. The phoncode corrector succeeds in correcting 'phonetic' errors; the editcost corrector corrects these and others in addition.
- (f) The children fail at different stages in the spelling process: in the splitting of speech sounds into phonemes; in the selection of plausible graphemes to represent these phonemes; and in the use of orthographic information to select the correct grapheme from the plausible ones. This failure at different stages is reflected in the performance of the correction programs in correcting their errors.
- (g) The correction programs could be extended to provide more information about the children's errors. They are also suitable for use with other groups of children.

## Appendix A

### Editing commands, stories and notes on the children

This appendix contains:

- a. a summary of commands for the Walter text editor;
- b. copies of stories for study 1 and study 2;
- c. details about the individual children in the two studies;
- d. the dribble file and corresponding writing from one child, for one study 2 session;

1. NEW - to write a new story

W: new

STORY: once upon a time there was a little

STORY: girl called Sally who lived all alone

STORY: in the woods.

STORY:

NEW FINISHED

2. SAVE - to save a story.

W: save

NAME OF MEMORY BANK: story1

SAVE FINISHED

3. RECALL - to get a story back from the computer's memory.

W: recall

NAME OF MEMORY BANK: story1

\*\*\*\*\*

Once upon a time there was a little girl called Sally who lived  
all alone in the woods.

\*\*\*\*\*

4. CHANGE - to make any changes in a story.

W: change

OLD WORDS: girl

NEW WORDS: boy

OLD WORDS:

\*\*\*\*\*

Once upon a time there was a little boy called Sally who lived  
all alone in the woods.

\*\*\*\*\*

5. MEMORIES - to see the names of all the stories saved

W: memories

YOUR MEMORIES ARE story1 test

6. PRINT - to print out a copy of a story

W: recall

NAME OF MEMORY BANK: story1 (shows story1)

W: print

PRINT FINISHED (sends story1 to printer)

**Figure A-1: Summary of Walter commands**

**Stories written by the children in  
Study 1: corrected versions**

**C.M.**

**Myself**

My name is Charles. I live at 181 commercial street. I have a dog called kim. At school I have a guinea pig called busby

**The Ghost.**

At the back of my house there is a castle that is haunted. It says that there is a ghost at night, it is to haunt. At night you can hear noises.

**The Train.**

At the front of my house there is a railway. At night the train came through my house and the train blew its horn.

**Turtle.**

At the university there is a turtle that can draw pictures. You push some buttons like forward 100 and backward 150. The turtle dropped the pen on the board then it starts to draw the letters

**G.Q.**

**Myself.**

My name is Gary Quinn I live at Broughton Road. I go to Broughton Primary school. My mother works as a designer. My father works as a shipwright. Paul my Brother is at School. I like playing rugby.

**The Swindle.**

Once upon a time there was Archeology dig. A Miss Wood was in charge Bannockburn. A Mr Peacock was helping her.

**Turtle.**

Today I drew a G the directions are forward hundred left one hundred and fifty left ninety forward hundred left ninety forward twenty left ninety forward ten. Then I made a Q the directions are forward hundred left ninety forward hundred left ninety forward hundred left ninety forward hundred left forty five forward twenty backwards forty. You go right 100, then forwards fifty then backwards fifty then left ninety, then forwards twenty five, backwards fifty

**Blackhand.**

One night a lady was sitting in her cottage, and she heard a scraping noise scrape, scrape, scrape. She waited until her husband came home and told him what had happened. He said I'll have a look in the morning. So in the morning he had a look. And he saw scratches on the panes of glass. So that night he stayed at home with his wife. And the black hand came that night, scrape, scrape, scrape, smash, he got his axe and, and ran after it. He hit it with his axe, but it moved on.

**J.M.****Hotel.**

My name is John. I live in a Hotel. The Hotel is in Portobello Portobello is a DUMP. I do not like the Hotel. My dog's name is PEPSI she is lovely. She is a Jack Russell.

**Turtle.**

I work on a computer it was very good. The name of computer is turtle it is funny. It has a round cover. You can move it all around the board. You have to type it out on a button box.

**The Wreck.**

I went on a boat to a wreck it was very dark. The boat was slanted. It was lying on the rock's.

**The Scream.**

I live in a mansion it was built in 1717 many people were killed. I get nightmares this is what the nightmares are all about. It is winter and it is very cold I always leave the window open. The butler opens the door and the man came in and gave the butler some money and shut the door and the man comes up the stair. He got to the top of the floor so I ran to my room. I was watching the man from my room the man was the doctor I was surprised a gasp come out of me. He pulled out a big sharp knife he went into my mum's room a squeak come from the floorboards and then a scream come from the room there was a big howl of wind I turned. The curtain wiggled and fell on the floor. I was going to scream but I never. So I went to my mum's room I saw blood but not my mum. I went to my room and I got my case and packed and I ran away and I never came back again

**L.B.****Leon (my cat).**

My cat is very bad it takes things out of the bucket steals fish from

plates and has very long claws and fights with other cats and does not like milk

### **The Roman.**

90 years ago in a lonely quarry in Middleton, as work began, some workers at the front were talking. Then someone said I wonder how the old fool got on with his ghost hunting. They got to bottom. Look over there. There was a body lay cut up. One of us said hes wearing a brass skirt. Hey hes a roman. They died. What about my dad. Look, here's a note. Join the legion at midnight. Then the body came together. It said hullo its me dad. Remember to be here to night.

### **Turtle.**

Left 90 left 60 backward right 120 forward 60 that is how you write y on turtle. It is a machine that can write and draws and helps you with maths.

### **M.W.**

### **Turtle.**

I left the Reading Unit with John and Gary. We walked past the museum to the department of artificial intelligence. We walked up the stairs to logo, where he is kept. I had a sheet to copy from, I wrote numbers like this backward 100 right 90 forward 50.

### **Football.**

It was nine o'clock in the morning. I was playing for the school football team. It was 6. 0 for us Mr Scott was pleased at us. That was a decider for the league also we're in the Hammy cup. I've scored 16 goals.

### **Witches.**

I saw a horror film, it was about witches they were casting spells on people. Like killing people. They did it on Friday nights at midnight all the time. One night they got a man and put needles in him till he died then they threw him in a fire it burned all night.

### **Goals.**

I can remember scoring against South Morningside. I scored 4 goals against them and I scored a penalty, I scored a diving header. The other team did not score any at all.

### **N.M.**

**Turtle.**

I came to the department of artificial intelligence and we went to the logo room and we learnt about turtle. It draws letter and shapes.

**Nicky.**

My name is nicholas. My dog's name is lucky. My cat's name is smoky. My cat gets in fights and he has a hurt paw. I take my dog for walks to the motorway and back.

**John (a ghost story).**

At night I heard sounds upstairs when no body was upstairs, and my nana lost her earring down the back of the fireplace and the next day the earring was in the drawer.

**Peter.**

My uncle had a baby. It is a he, 5 months old. His name is Peter. He is my cousin.

**The Zoo.**

I went to the zoo with Miss Watson's class and we went by bus to the zoo. When we got there we saw seals and we went to the pheasant. It opened its feathers and people were taking photos of it. We went to reptile house and saw reptiles then we went to the bears and saw a brown bear and we saw two white bears too. We saw birds and fishs and goats.

**S.S.****Horror.**

Today I am watching the television. I ask dad to watch the tv. It is on the tv at 7. 15. And dad asks to turn over. I ask to dad is it started. The hammer house of horror. Then I turn over to watch the news starting at 10 o'clock.

**Sue.**

My name is susan. I live at 321 easter road edinburgh my cat name is peach. She live at 321 easter road edinburgh. She plays in the backgreen

**Turtle.**

Today I had a shot of the turtle. I join letters to make y then I came to make a story then I get the bus to the school in the school I get sewing then dinner time.

Stories written by the children in study 2:  
examples of originals

M.A.

Robots are used in factories for many of the jobs <sup>which</sup> ~~which~~ are unpleasant, unhealthy, ~~boring~~ or costly.

When you receive your robot you will have to program it. there are ~~two alternatives~~ ~~alternatives~~ alternatives of how to program your robot. You can tell it by typing in words and descriptions which is a lot quicker and easier to make <sup>alternations</sup> ~~alternations~~ than ~~showing~~ it by moving the hands and telling it to remember. It is also dangerous if it out of order. Some of the jobs that robots can do today are putting things ~~together~~ together, welding, spraying paint and putting chocolates in boxes.

In the future it is hoped ~~that~~ <sup>that</sup> robots will do things faster and cheaper.

T.E.

Terry Dickson Jumper  
 Ten years old hit is 4 foot 3 in.  
 Wait 4 stop ~~trousers~~ and ~~the~~  
 and a ~~tie~~ ~~trousers~~ hair eyes are  
 blue ~~evening~~ Brown and shoes  
 green shoes

Clors off my trousers black

The ~~hell~~ ~~was~~ in hell

I went in to the ~~trousers~~ there  
 I ~~put~~ my hand ~~into~~ through  
 and a ~~ghost~~ and a skeleton  
 gurs ~~came~~ on my hand  
~~gods~~ David and Darren  
 and Francis. ~~come~~ the ~~police~~  
 the ~~police~~ were ~~puzzled~~ police  
 the ~~we~~ seen Francis run

## Notes on individuals.

### Study 1 - Group 1

This group was generally the "least able" of all four groups of children in the two studies. The teacher usually worked with children of roughly equal ability during any one session. She described this group as "hopeless cases" and was especially happy for them to take part in the study, but did not anticipate that they would achieve a great deal. The exception in this group was LB. He was more able than the other members but had been moved to the Reading Unit from his school because of severe behavioural problems. It was requested that he should participate in the first session to see what the others were doing. The investigator was warned that his behaviour in class was usually extremely disruptive. No problems had been encountered in the Reading Unit, however. As none were encountered by the investigator LB joined the group for the whole period of the study. He was in fact found to be quiet in the first few sessions, and co-operative and sociable thereafter.

From observation of the group and discussion with the Reading Unit teacher, informal assessment of ability of the children was made. SS and CM had the greatest difficulties in spelling and also in reading; NM had fewer difficulties but was still less able than the children in group 2. LB was on a par with the more able children in group 2. Note that this is a rough comparative judgement.

The group as a whole applied themselves to the task of writing and were the least distracted of the four groups. The following are summaries of informal notes made by the investigator whilst observing the children.

#### **S.S.**

Dysphasic. Some words indecipherable. Syntax not good. Works very hard, very keen. Missed two sessions, but this didn't appear to affect work in other sessions. Got on well with writing the stories but made lots of errors. Great difficulty in correcting errors. Reads back "what thinks said", not "what actually wrote". Had difficulty reading. Coped with using most commands - didn't use capitals or add to stories.

#### **C.M.**

Auditory perceptual problems. Mixes up sounds a lot – even gets his own name wrong. Works very well, very keen. Gets on with stories when left on his own. Will guess spellings. Lots of errors – sounds mixed up. Can't make sound-to-letter correspondence even when word spelt aloud. Can spot some spelling and syntax errors, but needs help to correct them. Sometimes forgets what a word was meant to be. O.K. on commands, except sometimes filename instead of command after prompt. Misspells commands. Sometimes uses backspace instead of delete.

#### **N.M.**

Enjoys working, very keen. Works well on his own. Also keen to help others (too keen sometimes). Getting better at guessing corrections – can usually spot errors fairly well, but difficulty correcting. Some letter confusions – less in later stages? Good grasp of commands – very good using 'memories', 'recall', 'print' (no prompting). Occasionally too many 'returns'.

#### **L.B.**

Very quick to pick things up. Works well, but very quiet. Boisterous, gregarious, to and from department – subdued whilst working. Asked more questions, more interaction during later sessions. Spelling not bad – perhaps more lack of practice and experience (and motivation) than psychological disorder. Will spell words aloud if asked to by others – more often right than wrong. Usually guesses spelling of commands (others look them up). Usually spots errors – good guesses at corrections. Also sometimes spots syntax errors. Commands O.K., except forgets 'return' after commands.

#### **Study 1 – Group 2**

This group was described by the teacher as one of the more able of the groups attending the Unit at the time of the study. All were due to move on to secondary school the following year. They were more able than the children in group 1, but more easily distracted. Both GQ and MW had been reported as having behaviour problems. In terms of spelling ability, JM and GQ were seen as the most able and MW with more difficulties. The reading level of this group appeared to be close to the expected ability for their age.

#### **G.Q.**

Distracted easily. Bright, but gets frustrated if things don't work as expected, or when he doesn't know what to do. Works well when working, interested in project, but put off if not succeeding - likes to get things right. Spelling - usually able to spot errors, and to make some corrections. Not very keen on doing this - needs prodding (doesn't have to read back over work in school?). Keen to write though, imaginative. Fine on commands; copes with 'change' with capitals, and with adding to a story using change (i.e. change last word to last word plus additional text). Interested in how the computer works - especially enjoyed playing with the 'turtle'.

#### **M.W.**

Works hard for short periods, but easily distracted. Unfortunately, most frequent victim of machine crashes - couldn't log on at all in one session. Gets very frustrated when things don't work. Not as hard-working as others. Guesses quite difficult spellings - reasonable success. Needs prompting to do changes, but can guess which words incorrect. Can use commands - changes capitals and punctuation. Needs some prodding to work, but produces reasonable work when tries.

#### **J.M.**

Very patient. Works very well. Very highly motivated. Will happily write lots - brought in story written in class to put on the machine and correct. Asks lots of questions about spellings. Will make guesses at which words are wrong, and reasonable guesses for corrections - not always correct. Difficulties in copying. Fine on commands, especially context for single changes, and adding to stories. Worked best of group 2.

### **Study 2**

#### **Group 1**

The group comprising FR, TE, DV, and DR were considered to be of moderate ability in comparison with other groups attending the Unit. They were generally more easily distracted than the children in either group in study 1. There was also some friction within the group, between FR and the others (particularly TE): this added to the distraction. In general, however, they were interested in the project (a newspaper) and worked well it. They were particularly excited about

the 'letter writing' and 'logo turtle' sessions. Of the group, TE was reported to be the brightest, but FR produced the best work. DR experienced difficulties in both spelling and reading, but less so than TE. DV had less spelling difficulties but also experienced reading difficulties.

#### **F.R.**

FR showed clearly the best spelling and writing ability of the group. He was fairly articulate but somewhat nervous. Produced a number of noticeable phonetic errors. Frequently asked for help but used the computer to help him when it was suggested he should do so. Liked to be seen to be 'working well' - had an argument in one session over who had written the most. Very 'keen to please'. Didn't socialise with the others. No real difficulty with using the computer though had a tendency to panic when asked to repeat a command after a mistype.

#### **D.R.**

Did not make so many errors in writing but tended to select easy words and use simpler sentence structures. Usually worked on his own and didn't ask for help: generally lost attention when he did not get the word he wanted. More limited vocabulary than TE or FR. Missed a large number of sessions due to school commitments. No difficulty using the computer (very good with logo turtle).

#### **T.E.**

Least able spelling but most ambitious in words attempted. Tendency to mix-up and omit sounds. Difficulty thinking 'what to write', tried to avoid writing generally - compensated verbally! Almost hyper-active. Keen to distract attention from writing by telling long and involved stories. Apparently very bright (very articulate) but teacher reported that it was very difficult to get him to do anything. Some difficulty 'reading back' his own stories. Made a good approximation to some difficult spellings. Often tried 'dictating back to him' what he wanted to say in writing.

#### **D.V.**

Less severe spelling difficulties than DR but problems with pronunciation. Fairly easily distracted (by TE). Tendency to ask for confirmation of words especially

when using lookup. If told "It's up to you to decide" he was able to do so. Some reading difficulties. Sociable. Keen to use the computer.

## **Group 2**

This was the group of most intelligent children attending the Unit at the time of the study. All had very high IQ's. Oldest group - aged 11 and 12 years. They worked on a project about the A.I. Department, which they all took very seriously. They usually started writing and checking as soon as they got into the terminal room. ST was reported to have the highest IQ of the 4. He had the most difficulty with writing including very poor handwriting. GR, however, had the most severe spelling difficulties of those in the group. DI and MA produced the best text in terms of content, vocabulary, style and spelling. All tended to wait while each word was being checked and were a little frustrated by the slowness of the system. The more extensive vocabulary of this group caused more words to be 'not found' in the dictionary.

### **M.A.**

Made notes before writing some pieces. Confident and well-motivated; articulate and sociable. Knew fairly accurately which words she had misspelt and also had some idea of possible corrections. No apparent reading disability. Worked hard at writing and improving text. Did not restrict what she wrote to what she could spell. Concerned to be able to use words she wanted - used the checker to check them. Tended, as DI, to check words before she wrote them.

### **S.T.**

Very articulate but less keen to write. Often stuck for ideas. Tendency to play with the computer. Familiar with computers. Sat and did nothing for parts of some sessions. Very immature handwriting and syntax in written text. In cases where he thought a word was correct but it was not offered, he tended to add it himself and then 'confirm it was correct' by re-checking it! Did proof-read and correct text for putting in the project publication. Most easily distracted of the group.

### **D.I.**

As MA tendency to check spelling before writing word. Also, able to detect

words spelt incorrectly. Quiet, slightly withdrawn. Did not ask for help often. Developed strategy of checking spelling by using lookup: if the word was correct it was confirmed by the definition; if it was incorrect or absent he used the spelling checker. He was using the program very much as an on-line dictionary. Apparently absent from school a lot due to health problems.

**G.R.**

Most difficulties of this group. Also, least articulate and least socially competent. Difficult to follow his text - tended to be a bit confused. Also found difficulty in translating directions with the logo turtle. Had to be prompted on words to check in proof-reading. Chatted to the investigator and others in the group whilst writing: perhaps wanting attention. No particular difficulties using the computer.

## An example dribble file and corresponding writing from F.R. for one study 2 session

### Dribble file

{Notes: spacing used to present the text on the screen has been removed here to reduce the length of the example}

Script started on Wed Jun 1 08:52:30 1983

```
% spellprog.o
type in filenames as requested - type "no" to stop
name of file to be used
horror dict
another file?
general dict
another file?
no
dictionary set up
dictionaries set up
weights assigned
```

Please type your first name and then press the RETURN button  
francis

Hello,francis  
Welcome to the spelling checker

```
w:check
What word do you want to check?
w:finges
```

Wait a minute while I check it  
It could be

```
things
finish
friends
find
```

If you want to look up the meaning of one of these words  
then type "lookup" after the w:  
w:checkfingers

Sorry, I cannot do that command  
I can check or look up a word  
w:check  
What word do you want to check?  
w:fingers

Wait a minute while I check it  
It could be

things  
finish  
fire  
first

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:lookup {lookup used by investigator  
to check if word in dictionary}

Which word do you want to look up in the dictionary?

w:fingers

fingers

= is not in the dictionary

w:check

What word do you want to check?

w:eyes

Wait a minute while I check it  
It could be

eyes  
yes  
easy  
eye

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:scelletn

Wait a minute while I check it  
It could be

skeleton  
called  
cellar  
collect

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:gostst

Wait a minute while I check it  
It could be

ghosts

ghost  
gets  
guts

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:bodys

Wait a minute while I check it  
It could be

body  
bodies  
bodyless  
boy

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:tery

Wait a minute while I check it  
It could be

terry  
try  
they  
there

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:bloud

Wait a minute while I check it  
It could be

blood  
bloody  
blue  
bolted

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:francis

Wait a minute while I check it  
It could be

francis  
friends  
front  
finish

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:murderd

Wait a minute while I check it  
It could be

murder  
murderer  
murderous  
more

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:rd:check

Sorry, I cannot do that command

I can check or look up a word

w:check

What word do you want to check?

w:througt

Wait a minute while I check it  
It could be

through  
thought  
though  
that

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:throgt

Wait a minute while I check it  
It could be

through  
thought  
though  
that

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:throught

Wait a minute while I check it

It could be

thought

through

though

fright

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:sore

Wait a minute while I check it

It could be

sure

some

so

corner

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:checkcheck

Sorry, I cannot do that command

I can check or look up a word

w:check

What word do you want to check?

w:sour

Wait a minute while I check it

It could be

sure

so

shout

sound

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check

What word do you want to check?

w:souw

Wait a minute while I check it

It could be

slow  
cow  
saw  
shout

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:check,  
Sorry, I cannot do that command  
I can check or look up a word  
w:check

What word do you want to check?  
w:opend

Wait a minute while I check it  
It could be

opened  
open  
opens  
opening

If you want to look up the meaning of one of these words  
then type "lookup" after the w:

w:goodbay

Sorry, I cannot do that command  
I can check or look up a word  
w:goodbye

Goodbye,francis

%

%

Original

I was walking along the  
 road looking for a house  
 when I ~~saw~~ <sup>saw</sup> lots of girls  
 and eyes looking at me  
 I be ~~gave~~ <sup>saw</sup> to run but then  
 I ~~saw~~ <sup>saw</sup> ~~skeleton~~ <sup>skeleton</sup> skeleton  
 in ~~front~~ <sup>front</sup> of me I ~~saw~~ <sup>saw</sup>  
 and then I fell then  
 I ~~saw~~ <sup>saw</sup> lots of ghosts  
 around me then I ~~saw~~ <sup>saw</sup>  
~~house~~ <sup>house</sup> and I rang the  
 bell then the door ~~opened~~ <sup>opened</sup>  
 then I ~~saw~~ <sup>saw</sup> guts it was  
 very smelly I ~~saw~~ <sup>saw</sup> dead  
~~bodies~~ <sup>bodies</sup> bodies so I called  
~~my~~ <sup>my</sup> came he ~~said~~ <sup>said</sup> he had  
~~found~~ <sup>found</sup> blood blood  
 we went up to the room  
~~where~~ <sup>where</sup> it was it was ~~very~~ <sup>very</sup>  
~~corrupted~~ <sup>corrupted</sup> the wall ~~sided~~ <sup>sided</sup>  
 my hand ~~went~~ <sup>went</sup> through  
 through

then we heard foot steps  
 then ~~very~~ <sup>very</sup> ~~he~~ <sup>he</sup> was  
~~murdered~~ <sup>murdered</sup> murdered then  
 was ~~blood~~ <sup>running</sup> running from his  
 face so I ran out ~~to~~ the  
 house up the ~~St~~ <sup>St</sup> ~~at~~ <sup>at</sup>  
 and called the police.

**Final Version**

I was walking along the road looking for a house when I saw lots of fingers and eyes looking at me. I began to run but then I saw a skeleton in front of me. I screamed and then I fell. Then I saw lots of ghosts around me. Then I saw a house and I rung the bell. Then the door opened. Then I saw guts, it was very smelly. I saw dead bodies so I called Terry . Terry came. He said he had found blood we went up to the room where it was found. I touched the wall. Suddenly my hand went through, then we heard foot steps. Then terry fell. He was murdered. There was blood running from his face. So I ran out of the house, up the street and called the police.

## Appendix B

### Phoncode grammar and examples

In this appendix the following are presented:

- (a) the grapheme-phoneme grammar used by the phoncode program;
- (b) a sample of dictionary words, phonetically coded;
- (c) example parses of words from the dictionary, according to the grammar.

In the actual program the phonemes are coded as integers. In this appendix they are represented with their character codes for ease of reading.

The full grapheme-phoneme grammar used by the phoncode program is given in figure B-1. Tied graphemes are indicated by a "1" preceding them and segmentable graphemes by a "0". All 'vc+e' graphemes are segmentable, and are indicated by "\*" (for a further explanation see section 7.3.5).

A sample of words from the phonetically coded dictionary is given in figure B-2.

Parses of four words from the sample dictionary are illustrated in figure B-3. These are parsed according to the phoncode grammar. The alternative representation is used, as described in section 7.3.8.

0augh => o: a:	0air => eE
leigh => eI aI	0aer => e:
0ough => o: aU u E EU	0are => eE
ligh => aI	0ar => a: r E
leau => EU ju	0ear => a: e: IE eE
0tch => ch	0ere => IE
lcs => ks	0re => e: r E
0sc => s	0er => e: r E
ldj => dz	0ier => IE
lbb => b	0ir => e: E r
lcc => k ks s	0our => o:
lck => k	0oor => o:
ldd => d	0ore => o:
lff => f th tv	0or => o: e: r E
lgg => g	0ure => e: r
lll => l	0ur => E e: r
lmm => m	*au => EU eI O o:
lnn => n	*eu => ju
lpp => p	*ou => aU
lrr => r	*ai => eI eE
0ssi => sh	*ie => i aI I IE
lss => s z zh sh	*ee => i IE
ltt => t	*ea => i IE eI eE e
lzz => z zh	*oi => oI
lch => k ch sh	*ei => i I
lgh => f g	lae => i eE E eI IE
lph => f p	laa => a:
lsh => sh s ch	lai => eI eE E aI I e
lth => th tv	lau => EU aU o: a: O
lwh => h w	lee => i IE eI e E
lrh => r	0eu => ju
0eou => E	loa => EU o: O E aU
lqu => kw	loo => u U ^ EU o: O aU
0gue => g	lou => E EU u U ^ aU o: ju
0gu => g	lay => eI eE aI e I i
0dge => dz	ley => eI i aI I
0dg => dz	luy => aI
0ge => dz g	0hoy => oI
0ce => s z	loy => oI
0ci => s sh	liy => aI I
0cq => kw	0ye => aI I
0ayor => eE	0yoo => ju
0oar => e: o:	0you => ju

**Figure B-1:** Grapheme-Phoneme Correspondences:  
Grammar used in Phoncode Program

0yo => ju	0ks => ks
0ao => O	0lf => f v
0ei => i IE aI I eE eI	0lm => m
0ea => a: i IE e e: eI eE E	0lk => k
0eo => IE i	0ld => d
0ia => aI eE	0an => n
0ie => i IE aI I eE u	0en => n
0iu => ju	0on => n
0oi => oI u	0se => s z
0oe => EU u ^	0mb => m
0ua => O	0nm => m
0ui => u ju oI I	0ng => ng
0ue => u ju	0si => sh zh
0uo => ju EU aU	0sw => s
0aw => o:	0ti => ch sh zh
0ouw => o:	0ct => ch
0iew => ju	0wr => r
0ew => u ju EU	la => eI eE ae a: e e: E O o: ^ I
0ow => EU aU o:	le => i IE e e: E I eI eE ae
0uw => u ju w	li => e: E aI I y i e ^
*y => aI I	lo=>e: E EU o: u U ^ O I aU ju w
*a => eI eE ae a: I E aI e	lu => e: E U ju ^ w u I e
*e => i IE e e: eE E	lb => b
*i => i aI I e E	lc => k s sh
*o => EU o: O ^ u oI	ld => d dz t
*u => u U ju ^ o: E ae	lf => f v th tv
0ps => s	lg => g zh dz k
0bt => t	lh => h
0ct => k	ll => l
0dt => d	lm => m
0al => o:	ln => n ng m
0el => l	lr => r e: E
0il => l	ls => s sh z zh
0ul => l	lt => t ch sh d th
0le => l	lp => p
0ed => t d	lq => kw k
0de => d	lj => dz
0es => z zh s	lk => k
0ve => v f	lv => v
0gn => n	lw => w
0gi => dz	lx => g k ks s z zh
0kn => n	ly => aI I y i ju
0kw => kw	lz => z zh s sh

Figure B-1, continued

add	ae d	hello	h e l EU
adjust	ae dz ^ s t	help	h e l p
adrift	E d r I f t	helped	h e l p d
adventure	ae d v e n ch E	kind	k a I n d
adventure	ae d v e n ch E r	kingston	k I n g s t E n
after	a: f t E r	kitchen	k I ch E n
after	a: f t E	knee	n i
again	E g e I n	knees	n i z
against	E g e I n s t	knew	n ju
age	e I dz	knife	n a I f
ago	E g EU	knight	n a I t
ahead	E h e d	knock	n O k
aims	e I m s	knocked	n O k d
air	e E r	know	n EU
alan	ae l E n	night	n a I t
alex	ae l e ks	nightmares	n a I t m e E z
alight	E l a I t	nightmares	n a I t m e E r z
all	o: l	nights	n a I t s
allowed	E l a U d	nine	n a I n
he	h i	ninety	n a I n t I
head	h e d	theory	th IE r I
headed	h e d I d	there	tv e E r
header	h e d E	these	tv i z
header	h e d E r	they	tv e I
hear	h IE	thicknesses	th I k n E s E z
hear	h IE r	wheel	w i l
heard	h e: d	wheels	w i l z
heard	h e: r d	when	w e n
height	h a I t	where	w e E r
held	h e l d	which	w I ch
		while	w a I l

**Figure B-2:** Examples of the phonetically coded dictionary

```

error=nee
node = 1
  phon = /n/ , next = 2
  phon = /ng/ , next = 2
  phon = /m/ , next = 2
node = 2
  phon = /eI/ , next = 4
  phon = /i/ , next = 4
  phon = /IE/ , next = 4
  phon = /e/ , next = 4
  phon = /E/ , next = 4
node =3
=knee

error=nite
node =1
  phon = /n/ , next = 2
  phon = /ng/ , next = 2
  phon = /m/ , next = 2
node =2
  phon = /i/ , next = 103
  phon = /i/ , next = 3
  phon = /e/ , next = 3
  phon = /e:/ , next = 3
  phon = /E/ , next = 3
  phon = /aI/ , next = 3
  phon = /aI/ , next = 103
  phon = /I/ , next = 3
  phon = /I/ , next = 103
  phon = /y/ , next = 3
  phon = /e/ , next = 103
  phon = /E/ , next = 103
  phon = /^/ , next = 3
node =3
  phon = /ch/ , next = 4
  phon = /sh/ , next = 4
  phon = /t/ , next = 4
  phon = /d/ , next = 4
  phon = /th/ , next = 4
node =4
  phon = /eI/ , next = 5
  phon = /eE/ , next = 5
  phon = /ae/ , next = 5
  phon = /i/ , next = 5
  phon = /IE/ , next = 5
  phon = /e/ , next = 5
  phon = /e:/ , next = 5
  phon = /E/ , next = 5
  phon = /I/ , next = 5
=knight
=night

error=thair
node = 1
  phon = /th/ , next = 3
  phon = /tv/ , next = 3
node = 2
node = 3
  phon = /eI/ , next = 5
  phon = /eE/ , next = 5
  phon = /eE/ , next = 6
  phon = /e/ , next = 5
  phon = /E/ , next = 5
  phon = /aI/ , next = 5
  phon = /I/ , next = 5
node =4
node =5
  phon = /e:/ , next = 6
  phon = /E/ , next = 6
  phon = /r/ , next = 6
=there

error=aymz
node =1
  phon = /eI/ , next = 3
  phon = /eE/ , next = 3
  phon = /e/ , next = 3
  phon = /aI/ , next = 3
  phon = /I/ , next = 3
  phon = /i/ , next = 3
node =2
node =3
  phon = /m/ , next = 4
node =4
  phon = /s/ , next = 5
  phon = /sh/ , next = 5
  phon = /z/ , next = 5
  phon = /zh/ , next = 5
=aims

```

**Figure B-3:** Examples parses of words according to phoncode grammar: alternative representation

## Appendix C

### Corpus of errors - both studies

#### STUDY 1 - ERRORS

##### GQ

shipwright	shipright	heard	herd
design	desin	scraping	sraping
rugby	ruby	scrape	srap
live	livea	scrape	scape
upon	apone	scratches	scaratches
miss	mis	scratches	scaretches
fifty	fity	panes	pains
ninety	ninty		

##### JM

hotel	hotle	money	monye
lovely	lovey	coming	comeing
jack	jacke	stair	stare
russell	rusul	man	mag
portobello	portobellw	surprise	supris
wreck	reck	pulled	pulld
rock	roack	knife	knif
computer	compuoter	squeak	squaek
computer	compouter	scream	skraem
turtle	turtul	scream	skream
funny	funy	there	thiar
cover	kuver	turned	ternd
board	bord	curtain	crtane
type	tipe	wiggled	wigule
built	bilt	change	cange
killed	kild	went	weat
nightmares	nightmars	seen	sean
very	verey	blood	blode
always	alwas	came	cam
watching	washing		

##### MW

won	one	witches	whitchs
mr	mir	witches	whitghs

scott	scot	witches	witchs
goals	gouls	were	where
left	lifet	spells	spels
john	jonh	needles	nedils
gary	garry	memories	memorys
stairs	stears	memories	memries
where	wear	threw	through
sheet	shet	a	are
numbers	numders	burned	berned
kept	cept	change	gh
kept	kapt	against	agenst
kept	kepit	diving	biving
artificial	artifishal	other	athrer
wrote	nrote	other	ather
wrote	krote	not	knot
goodbye	gooddy	goodbye	goodby

## LB

takes	takse	quarry	qorie
fight	fithse	middleton	midleton
steals	steels	began	bigan
plates	platse	talking	talkin
does	dose	fool	full
bucket	buket	ghost	gohst
fight	fihgts	there	thier
turtle	turtel	wearing	warin
machine	machein	skirt	skurt
machine	machien	hey	hay
can	cane	about	abuot
memories	memoris	legion	legon
lonely	lonly	hullo	hulo

## NM

fight	fihgt	turtle	tertle
paw	po	heard	haerd
paw	pau	department	department
take	taik	sounds	souns
hurt	hut	stairs	stares
motorway	mototorway	when	wen
change	caing	nana	nanana
change	chnge	lost	loast
nicky	nicy	cousin	coun
department	deppartment	month	munth
and	andt	cousin	cousen
logo	loggo	fireplace	fierplase
artificial	artifical	nicholas	nichoias
we	wer	watson	wautsan
learnt	lernt	taking	takeing
about	abowt	photos	photoso

fishes

fishs

CM

commercial

commrs

goodbye

goobay

commercial

commcil

buttons

buttns

charles

chaless

start

sart

street

steet

letters

lettes

called

colde

there

ther

called

collde

castle

castl

school

shool

that

thet

i

iy

haunted

hoted

guinea

giney

says

ses

busby

busbes

night

nairt

university

ynusty

haunt

hont

there

ther

you

yo

that

thet

hear

her

can

came

noises

noss

draw

droy

front

frunt

picture

picher

the

whe

push

posh

can

came

some

sum

train

tran

buttons

butns

going

golng

like

lake

through

thro

drop

drot

came

cane

board

bord

change

chage

charles

charess

blows

bols

recall

tecall

its

is

turtle

turle

horn

hon

change

calde

horn

hone

called

calde

ghost

gost

some

somb

hear

haes

some

sume

noise

nose

goodbye

gooboy

that

ther

goodbye

goodbay

ghost

ghoet

SS

easter

eastr

make

msea

road

rood

came

come

plays

pays

then

the

plays

pilas

get

cet

peach

paech

dinner

dinnrer

edinburgh

ediburgh

time

the

backgreen

daccgreen

memories

memoies

change

change

turtle

turle

new

neea

get

cedt

had

hat

change

chage

shot

shat

watching

waching

join

john

television

tahgfring

draw

john

tv

talhfi

make	mosea	turnover	turofr
horror	horey		

## STUDY 2 - ERRORS

FR

brown	broun	crowns	crouns
hair	hera	silver	isilver
hair	hare	looking	loking
clothes	cloths	saw	sore
eyes	irs	saw	sour
eyes	ias	saw	souw
eyes	ey	fingers	finiges
head	hard	skeleton	scelleten
head	herd	skeleton	scelletn
head	haid	skeleton	skelleten
shoes	shous	front	frount
shoes	shoues	screamed	scremd
chipped	chiped	screamed	screemed
two	to	ghosts	gosted
two	tow	ghosts	ghosted
years	yaers	ghosts	gostst
trousers	truusesrs	house	howse
goodbye	goodby	opened	opend
goodbye	goodbay	bodies	bodys
goodbye	goodbey	terry	tery
island	ilend	said	sead
forest	forist	blood	bloud
field	filed	where	warh
mountains	mountans	where	werh
treasure	tresher	found	fond
treasure	treasere	touched	tuched
diamonds	dimens	touched	tutched
jewels	jouls	suddenly	suddely
bodies	bodes	treasure	jouls
blood	blod	through	thow
computer	compter	screamed	screemed
procedure	prseeger	touched	tutched
chemicals	cemicls	using	uesing
photo	poto	magazine	magezean
scored	scord	week	weeke
jewels	treser	pictures	picters
machine	machien	pictures	pikters
went	wen	ribbons	ribons
through	troug	programmed	programed
through	trough	procedure	prseeger
through	thought	commands	comands
through	throgt	piece	pees

through	throght	piece	peces
murdered	murderd	piece	pieese
there	ther	piece	peesce
blood	plood	shapes	shaps
running	runing	there	thier
of	fo	chair	chive
done	dun	sofa	sofer
wembley	wembly	plans	plandes
players	playes	about	ubout
scottish	scotish	showed	sods
team	teame	showed	shode
macleish	maleish	showed	shodes
england	ingling	showed	sodes
england	ingland	took	toke
scored	scord	took	tooke
nicholas	nicolas	work	worke
write	rite	machine	machne
labour	laber	camera	camer
using	yoosing	picture	picter
does	dus	how	howe
magazine	magzine	photo	foto
magazine	magzeen	put	pit
magazine	magazien	develop	divelit
computers	compyooter	wyse	wizes
turtle	turtel	mine	mini
shell	sell	chemical	cemilkol
controls	contrals	chemical	cemikle
wheels	weels		

DV

years	yerse	want	wont
blue	blie	going	gon
eyes	liss	interview	intovue
eyes	isse	interview	interuien
brown	bloum	interview	intervien
brown	broum	interview	intervie
brown	broun	soldiers	soildde
brown	brouum	turtle	tultlue
brown	blounm	plastic	plasdid
hair	hear	plastic	plastid
hair	haer	plastic	plastad
wearing	waring	plastic	plaskid
jeans	jen	plastic	plaiked
bye	bi	wire	wiure
island	illing	does	does
island	iland	pictures	picte
of	over	pictures	pieces
of	ove	pictures	pictce
of	ov	help	hlep

of	ovre	stairs	stared
near	neer	stairs	stare
near	mear	down	domne
buried	berayd	down	dome
buried	beray	down	dame
ghost	goss	picture	pictur
house	houes	picture	picte
when	wen	picture	pictee
terry	tery	could	cood
darren	darin	could	cold
built	bulut	want	wont
cup	cap	gold	good
said	sed	working	woking
kenny	cenay	working	wuking
kenny	kenne	bit	bid
kenny	kenye	talk	tock
dalglish	dugle	drill	drule
dalglish	duglesh	more	mor
dalglish	dalglesh	david	davd
because	becose	francis	farasis
goals	gois	picture	pichur
should	shood	photo	fot
have	uve	photo	fotot
for	of	photo	front
goodbye	boodbye	white	withe
score	scorre	white	wite
scotland	scoltand	with	whith
magazine	magssen	francis	franses
magazine	magsen	knight	night
magazine	magen	thank	thack
years	yeas	ghost	gol
eyes	eye	house	houes
brown	brom	built	buld
hair	hard	little	littie
socks	sock	because	becoes
near	her	score	scor
treasure	teresur	should	shod
buried	berde	stairs	star
have	hath	plastic	palsick
lake	lack	want	wont
TE			
weight	wait	hops	hop
stone	ston	?jrle	jrle
trousers	troosers	want	wont
tie	tiy	write	right
brown	boron	downing	bownen
brown	bron	downing	downen
brown	broan	magazine	magasebn

blue	blule	conservative	kunsevrter
colours	clors	conservative	cunjnc
goodbye	goodbiy	conservative	sevter
traps	trips	conservative	konsevtatv
near	nerer	thatcher	thach
near	nere	thatcher	thacher
shipwreck	shipreg	mrs	missis
gold	golb	use	uoser
gold	goib	use	uose
nearer	nerer	draw	braw
david	daivd	make	maek
put	pot	instructions	instructins
there	ther	instructions	insrushn
back	dack	turtle	turtl
watch	woch	plastic	plasek
line	lifn	are	ar
line	lin	these	theser
hotel	honteb	these	theres
hotel	hote	called	colled
put	pot	found	foond
through	thro	cashline	caskline
through	thr	cashline	cashlin
ghost	gost	bare	bair
ghost	gosts	of	over
came	gam	of	off
came	come	house	hoses
out	ot	house	howes
called	golld	house	hos
called	cocled	where	were
called	colled	work	wrk
called	colld	line	lin
david	daved	any	eney
david	davet	any	eny
come	conn	any	ene
come	cone	tools	tols
police	polce	tools	toles
police	plec	the	ther
were	wer	metal	metel
puzzled	puzzel	metal	meetel
seen	cn	were	wer
seen	cen	very	vere
head	hoied	tough	thufh
skeleton	skeleton	would	wob
francis	franse	if	ifh
motor	mothir	took	toik
motor	moterdf	dark	barck
motor	moterd	we	wie
bike	dike	had	thad
wheel	weel	making	maken

rock	raock	developed	bvelupb
drive	brive	back	dack
dangerously	bandrie	terminal	termel
dangerously	bangersly	terminal	termer
dangerously	dangersly	david	davet
ireland	irlnd	david	daved
?cbigl	cbigl	just	tust
done	doo	walked	workt
bunny	dunny	took	tike
bunny	bune	our	or
bunny	buny	photo	foto
shoes	shose	police	plesa
shows	shos	put	poot
trousers	trosers	through	thro
brown	broen	motorbike	motrbike
hear	her	skid	skib
what	whot	bunny	buney
line	lin	dangerously	bangrielle
ghost	gost	conservative	gusertr
skeleton	skeletn	turtle	totel
david	davit	plastic	plastec
come	cume	photo	puot
DR			
ten	toen	stadium	stadeum
birthday	briday	scotland	scottland
birthday	brithday	people	peppla
two	to	brazil	bransl
have	haft	brazil	brasur
have	half	?par	par
trousers	tothers	?gondchanken	gondchankn
black	brack	turtle	turned
shirt	shurt	turtle	trener
light	lite	turtle	turend
yellow	yellow	of	off
are	ar	make	mark
goodbye	goodbiy	with	which
then	then	with	withh
island	inand	instructions	inchins
island	isand	instructions	incruns
pond	poond	instructions	chuns
the	then	instructions	intrunctons
cross	sross	instructions	instructons
swamps	swomps	instructions	instrutions
water	warer	instructions	inchuns
around	arond	instructions	instrunchins
crown	cwon	instructions	instrunton
crown	cuwon	picture	ping
check	pellrs	picture	picksher

are	ar	picture	pictshere
packed	park	picture	pishur
packed	parck	picture	picher
packed	par	picture	picter
packed	part	robot	romdt
packed	pakt	robot	rodert
england	enlland	robot	roder
england	aingland	robot	romdert
won	wun	robot	rodet
won	win	kitchen	kitshen
strachan	stacking	cook	ckook
strachan	cracking	cook	crook
nicholas	nickris	dead	beb
nicholas	nickis	dead	ded
unlucky	unluck	has	his
unlucky	unlunck	door	boor
lucky	luch	hall	hooll
took	toog	dog	bog
stadium	stamum	ghost	gost
stadium	stamun	carpet	kapit

GR

telephone	telephon	intelligence	intelegent
used	yosed	inelligence	intellegence
pictures	pitures	university	univesty
sharples	shovples	try	trie
teacher	teatcher	computer	computer
teacher	taecher	there	their
teacher	tatcher	wires	wiers
university	universta	boxes	boxs
called	could	even	eve
kingston	kinston	who	how
were	where	uses	yous
centres	centers	language	langue
their	there	about	aboat
they	the	perq	perk
tutor	tuter	wire	wier
sum	sume	microchip	mirchip
goes	gos	used	yoused
by	buy	puck	pirck
screen	scren	pattern	parteren
pressing	preason	pattern	pateren
straight	strat	write	wriet
able	abul	alphabet	alphapet
answered	answedd	commands	camands
answered	answerd	quit	quite
machine	machene	a	and
artificial	artfichel	put	pit
any	ena		

## DI

models	modles	controlled	controled
smaller	smaler	operate	opperate
their	there	by	buy
millimetres	millimeteres	typing	typeing
centimetres	centimeeters	check	checkk
centimetres	centimeters	stick	stik
needed	neaded	specifications	spesifications
instrument	instramen	specifications	spec
designed	designned	perq	perk
motors	moters	you	yoy
which	whish	terminal	ternimal

## MA

factories	factuares	putting	puting
factories	factures	wither	ether
suitable	sutable	easier	easier
what	wat	moving	moveing
description	discription	which	whitch
spraying	spairing	kind	kined
spraying	spaing	appearance	appaerance
welding	wealding	been	bean
unpleasant	unplesant	instructions	instrutions
hoped	hopped	procedure	proseador
which	whilch	perq	purk
which	whick	perq	pirck
boring	borring	perq	purq
boring	borning	using	useing
receive	resive	pencil	pensil
receive	receiv	paddle	padle
receive	resiev	somewhere	somewhare
alternatives	alteration	thicknesses	thiknesses
alterations	alternations	picture	pickture
alterations	alternion	else	els
order	oder	making	makeing

## ST

does	dose	tune	thune
buggy	bugy	procedures	prosegers
buggy	bugi	procedures	prgrame
computer	commputer	build	biuld
mathematics	mathamatics	computer	commputers
technic	tecnic	you	yo
government	goverment	add	asdd
motors	mottors	spaces	spaceses
controlling	conttrolling	patterns	patearns
peter	peper	patterns	pattrens
turtle	tutle	copy	copyy
using	useing	another	anther

commands	comands	three	theree
numeric	numerical	been	beny
angle	angel	coming	comming
draw	drow	here	herar
programmes	progames	here	heer
so	sow	possible	posibale
have	huve	possible	posiblle
instructions	instruckshons	possible	possabile
tune	chun	perq	perc

## References

- (Alberga, 1967) Alberga, C.N.  
String Similarity and Misspellings.  
*Communications of the ACM* 10(5):302-313, May, 1967.
- (Albrow, 1974) Albrow, K.  
The Nature of the Writing System and its Relation to Speech.  
In Wade, B. and Wedell, K. (editors), *Spelling: Task and Learner*,  
University of Birmingham, Birmingham, 1974.
- (Allen, 1981) Allen, J.  
Linguistic Based Algorithms Offer Practical Text-to-Speech  
Systems.  
*Speech Technology* 1(1), fall, 1981.
- (Atkinson, 1974) Atkinson, R.C.  
Teaching Children to Read Using a Computer.  
*American Psychologist* 29:169-178, 1974.
- (Attisha and Yazdani, 1983)  
Attisha, M. and Yazdani, M.  
A Micro-Computer Based Tutor for Teaching Arithmetic Skills.  
*Instructional Science* 12:333-342, 1983.
- (Atwell, 1983) Atwell, E.  
Constituent-likelihood Grammars.  
*ICAME News: newsletter of the International Computer Archive  
of Modern English* 7:34-67, May, 1983.
- (Backhouse, 1979)  
Backhouse, R.  
*The Syntax of Programming Languages: Theory and Practice*.  
Prentice-Hall International, London, 1979.
- (Baker, 1980) Baker, R.G.  
Orthographic Awareness.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
3, pages 51-68. Academic Press, London, 1980.
- (Baron, 1977) Baron, J.  
Mechanisms for Pronouncing Printed Words: Use and Acquisition.  
In Laberge, D., Samuels, S.J. (editor), *Basic processes in reading-  
ing. Perception and Comprehension*, chapter 6, pages  
175-216. Lawrence Erlbaum, Hillsdale, NJ., 1977.
- (Barron, 1980) Barron, R.W.  
Visual and Phonological Strategies in Reading and Spelling.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
9, pages 195-213. Academic Press, London, 1980.

- (Bennett, 1967) Bennett, D.M.  
*New Methods and Materials in Spelling.*  
Australian Council for Educational Research. Dominion Press,  
Victoria, Australia, 1967.
- (Blair, 1960) Blair, C.R.  
A Program for Correcting Spelling Errors.  
*Information and Control* 3:60-67, 1960.
- (Block, 1979) Block, K.K.  
Cognitive Theory, CAI and Spelling Improvement.  
*Journal of Computer-Based Instruction* 5(4):86-95, May, 1979.
- (Brown and VanLehn, 1980)  
Brown, J.S. and VanLehn, K.  
Repair Theory: A Generative Theory of Bugs in Procedural  
Skills.  
*Cognitive Science* 4:379-426, 1980.
- (Bruce and Rubin, 1984)  
Bruce, B. and Rubin, A.  
QUILL Project.  
In *International Conference on Education and New Information  
Technologies*, . Center for Educational Research and In-  
novation, Paris, 1984.  
Summary.
- (Brueckner and Bond, 1955)  
Brueckner, L.J. and Bond, G.L.  
*Diagnosis and Treatment of Spelling Difficulties.*  
Appleton-Century-Crofts, Inc., New York, 1955, pages  
346-380, chapter 11.
- (Bryant and Bradley, 1980)  
Bryant, P.E. and Bradley, L.  
Why Children sometimes Write Words which they do not Read.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
16, pages 355-370. Academic Press, London, 1980.
- (Burton, 1982) Burton, R.R.  
DEBUGGY: Diagnosis of Errors in Basic Mathematical Skills.  
In Sleeman, D. and Brown, J.S. (editors), *Intelligent Tutoring  
Systems*, . Academic Press, London, 1982.
- (Cahen et al, 1971)  
Cahen, L.S., Craun, M.J., and Johnson, S.K.  
Spelling Difficulty - a Survey of the Research.  
*Review of Educational Research* 41(4):281-301, 1971.
- (Candy, 1982) Candy, L.  
A Study in the Use of a Computer as an Aid to English Teach-  
ing.  
*International Journal of Man-Machine Studies* 16:333-339,  
1982.

- (Cassidy, 1984) Cassidy, S.  
The Use of a Speech Synthesizer in the Teaching of Reading.  
Master's thesis, Department of Artificial Intelligence, Sept.,  
1984.
- (Chomsky and Halle, 1968)  
Chomsky, N. and Halle, M.  
*The Sound Pattern of English*.  
Harper and Row, New York, 1968.
- (Ciarcia, 1982) Ciarcia, S.  
Build the Microvox Text-to-Speech Synthesizer. Part 2:  
Software.  
*BYTE* 40-64, October, 1982.
- (Clarke, 1982) Clarke, M.  
Talk presented at U.K.R.A. conference, and personal communication.
- (Cohen, 1980) Cohen, G.  
Reading and Searching for Spelling Errors.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
7, pages 135-155. Academic Press, London, 1980.
- (Collins, 1982) Collins, A.  
Teaching Reading and Writing with Personal Computers.  
1982.  
Written under contract for the National Institute of Educational  
Reading Synthesis Project.
- (Cornew, 1968) Cornew, R.W.  
A Statistical Method for Spelling Correction.  
*Information and Control* 12, 1968.
- (Cumming et al, 1984)  
Cumming, G.D., Richards, T. and Woodhouse, D.  
Use of Computers in Teaching Beginning Reading.  
In *International Conference on Education and New Information  
Technologies*, . Center for Educational Research and In-  
novation, Paris, 1984.  
Summary.
- (Damerau, 1964) Damerau, F.  
A Technique for Computer Detection and Correction of Spell-  
ing Errors.  
*Communications of the ACM* 7(3):171-176, March, 1964.
- (Davidson, 1962) Davidson, L.  
Retrieval of Misspelled Names in an Airlines Passenger Record  
System.  
*Communications of the ACM* 5:169-171, March, 1962.

- (Desberg, 1980) Desberg, P., Elliott, D.E. and Marsh, G.  
American Black English and Spelling.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
4, pages 69-82. Academic Press, London, 1980.
- (Durham et al, 1983)  
Durham, I., Lamb, D.A., Saxe, J.B.  
Spelling Correction in User Interfaces.  
*Communications of the ACM* 26(10):764-773, October, 1983.
- (Ehri, 1980) Ehri, L.C.  
The Development of Orthographic Images.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
14, pages 311-338. Academic Press, London, 1980.
- (Ellis, 1981) Ellis, N.C.  
Project Application.  
Personal communication.
- (Ellis and Miles, 1977)  
Ellis, N.C., Miles, T.R.  
Dyslexia as a Limitation in the Ability to Process Information.  
*Bulletin of the Orton Society* 27:72-81, 1977.
- (Ellis and Miles, 1978a)  
Ellis, N.C., Miles, T.R.  
Visual Information Processing as a Determinant of Reading  
Speed.  
*Research in Reading* 1(2):108-120, 1978.
- (Ellis and Miles, 1978b)  
Ellis, N.G., Miles, T.R.  
Visual Information Processing in Dyslexic Children.  
In Gruneberg, Morris and Sykes (editor), *Practical Aspects of  
Memory*, pages 561-569. Academic Press, 1978.
- (Ellovitz et al, 1976)  
Ellovitz, H.S., Johnson, R.W., McHugh, A., Shore, J.E.  
Automatic Translation of English Text to Phonetics by Means  
of Letter to Sound Rules.  
*IEEE Transactions on Acoustics, Speech and Signal Processing*  
24:446, 1976.
- (Farnham-Diggory and Simon, 1975)  
Farnham-Diggory, K. and Simon, H.A.  
Retention of Visually Presented Information in Children's Spell-  
ing.  
*Memory and Cognition* 3(6):599-608, 1975.
- (Fass, 1983) Fass, D. C.  
The Design of a Spelling Corrector Using Context.  
Master's thesis, Dept. of Computer Science, University of Es-  
sex, April, 1983.

- (Fass, 1984) Fass, D.  
Personal communication: Ph.D. thesis in progress based on M.Sc. work.
- (Fass and Wilks, 1983) Fass, D., Wilks, Y.  
Preference Semantics, Ill-formedness, and Metaphor.  
*American Journal of Computational Linguistics* 9(3-4), July-December, 1983.
- (Fiddy, 1982) Fiddy, P.  
Microcomputers: Allies to Infant Teachers.  
1982.  
Unpublished Paper.
- (Fremer and Anastasio, 1969) Fremer, J. and Anastasio, E.J.  
Computer Assisted Item Writing - I: Spelling Items.  
*Journal of Educational Measurement* 6(2):69-74, 1969.
- (Frith, 1978) Frith, U.  
Spelling Difficulties.  
*Journal of Child Psychology and Psychiatry* 19(3):279-285, July, 1978.
- (Frith, 1980a) Frith, U. (editor).  
*Cognitive Processes in Spelling*.  
Academic Press, London, 1980.
- (Frith, 1980b) Frith, U.  
Unexpected Spelling Problems.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter 22, pages 495-515. Academic Press, London, 1980.
- (Galli and Yamada, 1968) Galli, E.J., Yamada, H.M.  
Experimental Studies of Computer-Assisted Correction of Orthographic Text.  
*IEEE Transactions on Engineering Writing and Speech* 11(2):75-84, 1968.
- (Gentry, 1978) Gentry, J.R.  
Early Spelling Strategies.  
*The Elementary School Journal* 79(2):88-92, November, 1978.
- (Gibson, 1969) Gibson, S.K.  
Computer Aided Construction of Categorized Spelling Tests.  
ERIC: ED 039 226.  
University of Washington, Bureau of Testing, Report June 1969.

- (Grocke, 1982) Grocke, M.A.  
Interactive Development of Reading Skills in an Educational Clinic.  
1982.  
Paper Presented at 18th Nat. Conf. of Australian Group for the Scientific Study of Mental Deficiency.
- (Groff, 1977) Groff, P.  
*Phonics, How and Why*.  
General Learning Press, New Jersey, U.S.A., 1977.
- (Guthrie and Siefert, 1977)  
Guthrie, J.T., Siefert, M.  
Letter-Sound Complexity in Learning to Identify Words.  
*Journal of Educational Psychology* 69(6):686-696, 1977.
- (Hammill, 1975) Hammill, D.D., Noone, J.  
Improving Spelling Skills.  
In Hammill and Bartell (editor), *Teaching Children with Learning and Behaviour Problems*, chapter 4. Allyn and Bacon Inc., Boston, 1975.
- (Hanna et al, 1966)  
Hanna, P.R., Hanna, J.S., Hodges, R.E., Rudorf, E.H.  
*Phoneme-Grapheme Correspondences as Cues to Spelling Improvement*.  
US Office of Education, Washington, 1966.
- (Harpin, 1976) Harpin, W.  
*The Second 'R'*.  
Unwin Educational Books, London, 1976.
- (Hasselbring and Crossland, 1982)  
Hasselbring, T.S. and Crossland, C.L.  
Application of Microcomputer Technology and Spelling Assessment of Learning Disabled Students.  
*Learning Disability Quarterly* 5(Winter):80-82, 1982.
- (Henderson and Chard, 1980)  
Henderson, L. and Chard, J.  
The Reader's Implicit Knowledge of Orthographic Structure.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter 5, pages 85-116. Academic Press, London, 1980.
- (Hicks, 1980) Hicks, C.  
Modality Preference and the Teaching of Reading and Spelling to Dyslexic Children.  
*British Educational Research Journal* 6(2), 1980.
- (Hildreth, 1955) Hildreth, G.  
Entry on Spelling.  
In *Encyclopedia of Education*, pages 387. MacMillan and Free Press, 1955.

- (Horn, 1969)      Horn, E.  
Entry on Spelling.  
In American Educational Research Association (editor),  
*Encyclopedia of Educational Research*, pages 1282-1293.  
MacMillan Co., London, 1969.
- (Hornsby and Shear, 1978)      Hornsby, B. and Shear, F.  
*Alpha to Omega: The A-Z of Teaching Reading, Writing and Spelling.*  
Heinemann Educational Books, London, 1978.  
Second Edition.
- (Howe, 1979)      Howe, J.A.M.  
*Some roles for the computer in special education.*  
DAI Research Report 126, Department of Artificial Intelligence,  
University of Edinburgh, 1979.
- (Howe et al, 1978)      Howe, J.A.M., Beattie, A., Cassels, F., Johnson, J. and Anderson, A.  
*Teaching Handicapped Children to Read in a Computer-Based Learning Environment*  
DAI Research Report 57, Department of Artificial Intelligence,  
University of Edinburgh, 1978.
- (Hughes, 1973)      Hughes, J.M.  
*Phonics and the Teaching of Reading.*  
Evan Brothers Ltd., London, 1973.
- (Jung, 1980)      Jung, U.  
Reading, Writing and Phonology in Modern Foreign-Language Teaching.  
In Felix, S.W. (editor), *Second Language Development. Trends and Issues*, pages 249-263. Tubingen, 1980.
- (Leveux, 1977)      Leveux, S.A.  
Computer Assisted Spelling and Arithmetic.  
*Computer Education* 26:25, June, 1977.
- (Lewis, 1980)      Lewis, M.  
*Improving SOLO's User-Interface: an Empirical Study of User Behaviour and a Proposal for Cost Effective Enhancements*  
Technical Report, The Open University, Milton, Keynes, 1980.  
Unpublished draft.
- (Macdonald, Frase et al., 1982)      Macdonald, N.H., Frase, L.T., Gingrich, P.S., and Keenan, S.A.  
The Writers Workbench: Computer Aids for Text Analysis.  
*IEEE Transactions on Communications*, January, 1982.  
Special Issue on Communications in the Automated Office.

- (MacKay, Thompson and Schaub, 1978)  
 MacKay, D. Thompson, B. and Schaub, P.  
*Breakthrough to Literacy Teacher's Manual*  
 Illustrated edition, London, 1978.
- (Malmquist, 1973) Malmquist, E.J.  
 Retardation.  
 In Staiger, R.C. (editor), *The Teaching of Reading*, chapter 6.  
 Ginn & Co., Unesco, Paris, 1973.
- (Marsh et al., 1980)  
 Marsh, G., Friedman, M., Welch, V. and Desberg, P.  
 The Development of Strategies in Spelling.  
 In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
 15, pages 339-353. Academic Press, London, 1980.
- (Masters, 1927) Masters, H.V.  
*A Study of Spelling Errors*  
 PhD thesis, University of Iowa, 1927.  
 Unpublished.
- (McMahon et al, 1978)  
 McMahon, L.E., Cherry, L.L., Morris, R.  
 Statistical Text Processing.  
*Bell System Technical Journal* 57(6):2137-2154, July-August,  
 1978.
- (Miles and Wheeler, 1974)  
 Miles, T.R., Wheeler, T.J.  
 Towards a New Theory of Dyslexia.  
*Dyslexia Review* 10:9-11, 1974.
- (Mitton, 1984a) Mitton, R.  
 Personal Communication.
- (Mitton, 1984b) Mitton, R.  
*A Corpus of Misspellings Analysed by Simple Error-Types and  
 by Soundex Codes*  
 Research Report JRM/06/84, Dept. of Computer Science,  
 Birkbeck College, University of London, 1984.
- (Mor and Fraenkel, 1982)  
 Mor, M., Fraenkel, A.S.  
 A Hash Code Method for Detecting and Correcting Spelling Er-  
 rors.  
*Communications of the ACM* 25(12):935-938, December, 1982.
- (Morgan, 1970) Morgan, H.L.  
 Spelling Correction in Systems Programs.  
*Communications of the ACM* 13(2):90-94, 1970.

- (Morris-Wilson, 1984) Morris-Wilson, I.  
*English Phonemic Transcription.*  
Basil Blackwell Ltd., Oxford, 1984.
- (Munnecke, 1980) Munnecke, T.  
Give your Computer an Ear for Names.  
*BYTE*:196-200, May, 1980.
- (Naidoo, 1972) Naidoo, S.  
*Specific Dyslexia - the Research Report of the Invalid Children's Aid Association Word Blind Centre for Dyslexic Children.*  
Pitman Press, London, 1972.
- (Nelson, 1980) Nelson, H.  
Analysis of Spelling Errors in Normal and Dyslexic Children.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter 21, pages 475-493. Academic Press, London, 1980.
- (Nolen, 1980) Nolen, P.  
Sound Reasoning in Spelling.  
*The Reading Teacher* 33(5):538-543, Feb., 1980.
- (O'Shea and Young, 1978) O'Shea, T. and Young, R.M.  
A Production Rule Account of Error's in Children's Subtraction.  
In *Proceedings of the AISB-78 Conference on Artificial Intelligence, Hamburg*, pages 229-237. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1978.
- (Omond, 1977) Omond, R.A.  
*A Spelling Correction System (the Speller).*  
Undergraduate Project Report, Department of Artificial Intelligence, University of Edinburgh, 1977.
- (Otto et al, 1973) Otto, W., McMenemy, R.A., Smith, D.J.  
Spelling.  
In *Corrective and Remedial Spelling*, chapter 10. Houghton Mifflin, 1973.
- (Pain, 1980) Pain, H.G.  
*Analysis of Errors in Spelling*  
DAI Working Paper 61, Department of Artificial Intelligence, University of Edinburgh, 1980.
- (Pavlidis, 1979) Pavlidis, G.T.  
How can Dyslexia be Objectively Diagnosed?  
*Reading* 13(3):3-15, December, 1979.

- (Peters, 1974) Peters, M.L.  
The Significance of Spelling Miscues.  
In Wade, B. and Wedell, K. (editors), *Spelling: Task and Learner*,  
pages 56-60. University of Birmingham, Birmingham, 1974.
- (Peters, 1975) Peters, M.L.  
*Diagnostic and Remedial Spelling Manual*  
London, 1975.
- (Peterson, 1980a) Peterson, J.L.  
Computer Programs for Detecting and Correcting Spelling Errors.  
*Communications of the ACM* 23(12):676-687, December, 1980.
- (Peterson, 1980b) Peterson, J.L.  
*Lecture Notes in Computer Science* Volume 96: *Computer Programs for Spelling Correction: an experiment in program design*.  
Springer Verlag, 1980.
- (Pollock, 1982) Pollock, J.J.  
Spelling Error Detection and Correction by Computer: some notes and a bibliography.  
*Journal of Documentation* 38(4):282-291, December, 1982.
- (Pollock and Zamora, 1984)  
Pollock, J.J., Zamora, A.  
Automatic Spelling Correction in Scientific and Scholarly Text.  
*Communications of the ACM* 27(4):358-368, April, 1984.
- (Prinz and Nelson, 1984)  
Prinz, P.M. and Nelson, K.E.  
Child-Adult-Teacher Interactive Method for Teaching Reading and Writing to Young Deaf Children.  
In *International Conference on Education and New Information Technologies*. Center for Educational Research and Innovation, Paris, 1984.  
Summary.
- (Quinsaas, Levin and Gentner, 1980)  
Quinsaas, M.G., Levin, J.A. and Gentner, D.  
A Writer's Assistant: The Use of Personal Computers to Teach Writing.  
1980.  
Unpublished paper.
- (Sharples, 1984) Sharples, M.  
*Cognition, Computers and Creative Writing*  
PhD thesis, Department of Artificial Intelligence, 1984.
- (Simon, 1976) Simon D.P.  
Spelling. A Task Analysis.  
*Institutional Science* 5:277-302, July, 1976.

- (Simon and Simon, 1973)  
Simon, D.P., Simon, H.A.  
Alternative Uses of Phonemic Information in Spelling.  
*Review of Educational Research* 43:115-137, 1973.
- (Skousen, 1982) Skousen, R.  
English Spelling and Phonemic Representation.  
*Visible Language* 16(1):28-38, 1982.
- (Sloboda, 1980) Sloboda, J.A.  
Visual Imagery and Individual Differences.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
11, pages 231-248. Academic Press, London, 1980.
- (Smith, P.T., 1980)  
Smith, P.T.  
Linguistic Information in Spelling.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
2, pages 33-49. Academic Press, London, 1980.
- (Spache, 1940) Spache, G.  
A Critical Analysis of Various Methods of Classifying Spelling  
Errors, I.  
*Journal of Educational Psychology* 31:111-134, 1940.
- (Stevenson, 1984)  
Stevenson, H.W.  
Orthography and Reading Disabilities.  
*Journal of Learning Disabilities* 17(5), May, 1984.
- (Teitelman, 1978) Teitelman, W.  
*Interlisp reference manual*  
Xerox, PARC, California USA., 1978.  
Chapter 17.
- (Tenczar and Golden, 1962)  
Tenczar, P.J., Golden, W.W.  
*Spelling, Word and Concept Recognition*.  
Technical report CERL-X-35, University of Illinois, October,  
1962.
- (Tenney, 1980) Tenney, Y.J.  
Visual Factors in Spelling.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter  
10, pages 215-229. Academic Press, London, 1980.
- (Terrell, 1984) Terrell, C.D.  
Remedial Reading: CAL Programs and Speech Generation.  
1984.  
Draft, Unpublished Paper.
- (Valtin, 1978) Valtin, R.  
Dyslexia: Deficit in Reading or Deficit in Research.  
*Reading Research Quarterly* 19(2):201-221, 1978.

- (Venezky, 1966) Venezky, R.L.  
Automatic Spelling to Sound Conversion.  
In Garvin, P.L. and Spolsky, B. (editors), *Computation in Linguistics*, chapter 6. Indiana University Press, 1966.
- (Venezky, 1967) Venezky, R.L.  
English Orthography: its Graphical Structure and its Relation to Sound.  
*Reading Research Quarterly* X(3):75-105, 1967.
- (Venezky, 1980) Venezky, R.L.  
From Webster to Rice to Roosevelt: the formative years for spelling instruction and spelling reform in the U.S.A.  
In Frith, U. (editor), *Cognitive Processes in Spelling*, chapter 1, pages 9-30. Academic Press, London, 1980.
- (Wade and Wedell, 1974)  
Wade, B. and Wedell, K.  
*Educational Review Occasional Publications*. Volume 5:  
*Spelling: Task and Learner*.  
University of Birmingham, Birmingham, 1974.
- (Wagner and Fisher, 1974)  
Wagner, R.A. and Fisher, M.J.  
The String-to-String Correction Problem.  
*Journal of the ACM* 21(1):168-173, January, 1974.
- (Whiting and Jarrico, 1980)  
Whiting, S.A. and Jarrico, S.  
Spelling Patterns of Normal Readers.  
*Journal of Learning Disabilities* 13(1), Jan, 1980.
- (Wight-Boycott, 1977)  
Wight-Boycott, V.  
Fun and First Aid for Weak Spellers.  
*Remedial Education* :100-103, 1977.  
Cambridge Reading Project.
- (Wilby, 1980) Wilby, J.F.  
Jack Straw Reading Project.  
*Computer Education* :15-17, 1980.
- (Winograd, 1983) Winograd, T.  
*Language as a Cognitive Process: Syntax*.  
Addison-Wesley, 1983.
- (Yannakoudakis and Fawthrop, 1983a)  
Yannakoudakis, E.J. and Fawthrop, D.  
The Rules of Spelling Errors.  
*Information Processing and Management* 19(2):87-99, 1983.

(Yannakoudakis and Fawthrop, 1983b)

Yannakoudakis, E.J. and Fawthrop, D.

An Intelligent Spelling Error Corrector.

*Information Processing and Management* 19(2):101-108, 1983.

(Zamora, Pollock and Zamora, 1981)

Zamora, E.M., Pollock, J. and Zamora, A.

The Use of Trigram Analysis for Spelling Error Detection.

*Information Processing and Management* 17(6):305-316, 1981.

(Zinn, 1978)

Zinn, K.L.

Speak and Spell Spelling B.

*Creative Computing* :60-61, Sept/Oct, 1978.