Mitton, Roger (1996). *English spelling and the computer.* Harlow, Essex: Longman Group.

ENGLISH SPELLING AND THE COMPUTER

ROGER MITTON

Longman
London & New York

Contents

Publisher's acknowledgements

We are indebted to the following for permission to reproduce copyright material:

the author, David Holbrook and Cambridge University Press for samples of young people's writing in Appendix Three from his *English for the Rejected* (CUP, 1964);
the author, Dr P.D. McLeod for figure 6.2 based on a diagram from 'Overlapping mental operations in serial performance with preview: typing – a reply to Pashler' by P. McLeod and M. Hume in *Quarterly Journal of Experimental Psychology* **47A** (1): 193-9, 1994;
Pergamon Press Ltd for Chapter Four 'A Corpus of Spelling Errors' based on the article 'Spelling checkers, spelling correctors and the misspellings of poor spellers' by Roger Mitton in *Information Processing and Management* **23** (5): 495-505, 1987.

Author's acknowledgements

This book is based on the research I carried out for my PhD in Computer Science as a part-time student at Birkbeck College, London University. My first thanks are therefore to my PhD supervisor, Dr Trevor Fenner. A mathematician supervising a lapsed social scientist might have been tempted to take only a perfunctory interest in the work, but, on the contrary, he applied his mind to it with some enthusiasm. He is a sharp critic and a prolific generator of bright ideas and I am grateful for his contributions.

Others of my colleagues have also given assistance at various times, particularly Jim Inglis on programming and my former office companions Richard Murphy and Nigel Martin. From time to time I have inflicted spelling tests or spelling-related guessing games on various groups of people, notably the staff and students of the Department of Computer Science, and these have been borne with mostly good humour.

Early in the project I had a number of collections of spelling errors keyed into the computer, which involved a good deal of clerical work and data input. Philip Baker did much of this work and also, at a later stage, gave advice on linguistics. Others who contributed to this part of the project were John and Kate Murray and the staff of the Birkbeck Data Preparation Service – Lin Bailey, Sheila Hailey and Barbara Whitmore. This collecting of spelling errors was partly funded by a grant from the University of London Central Research Fund.

My thanks go to the Leverhulme Trust who gave a grant to Birkbeck College enabling me to work on the project full-time for

two years. A large part of this time was spent putting the machine-readable text of a published dictionary into a form suitable for my spelling corrector. People who helped with data entry or proofreading of my derived dictionary include Sylvia Davidson, Susan Drew, Ed Hastings, Ann Jones and Diana Whitaker.

For help with the test passages I am grateful to Anne Brewin, Ian Collett, David Holbrook and the students of Cassio College.

Uta Frith and Roger Garside, my PhD examiners, first encouraged me to turn my thesis into a book for publication.

Jim Inglis, Geoff Leech, Morag Stuart, Mick Short, Chris Upward and Peter Willmott kindly read drafts of the book or parts of it and gave me their comments.

Phil Docking, Mick Farmer and Keith Mannock have given me help with the text-formatting program troff and with the Postscript language that I used for creating the phonetic and other special characters.

Phonetic symbols used in the text

Consonants

p	in *pig*	s	in *sap*	ŋ	*ng* in *sing*
b	*big*	z	*zap*	θ	*th* in *thin*
t	*tog*	m	*map*	ð	*th* in *then*
d	*dog*	n	*nap*	ʃ	*sh* in *cash*
k	*hiker*	r	*rag*	ʒ	*ge* in *beige*
g	*tiger*	l	*lag*	tʃ	*tch* in *etch*
f	*fan*	w	*wag*	dʒ	*dge* in *edge*
v	*van*	h	*hag*	j	*y* in *yard*

Vowels

iː	in *bead*	eɪ	*day*
ɪ	*bid*	aɪ	*eye*
e	*bed*	ɔɪ	*boy*
æ	*bad*	əʊ	*go*
ɑː	*bard*	aʊ	*cow*
ɒ	*cod*	ɪə	*beer*
ɔː	*cord*	eə	*bare*
ʊ	*good*	ʊə	*tour*
uː	*food*		
ʌ	*bud*		
ɜː	*bird*		
ə	*a* in *ago*		

The symbols are taken from the International Phonetic Alphabet. The consonants are mostly represented by letters from the Roman alphabet and nearly all of these (the first two columns in the above table) have the values that an English reader would expect. The exception is /j/ which has the sound of *y* in *yellow* and not *j* in *junk* (*j* in *junk* is represented by dʒ). The Roman alphabet does not have enough vowel letters for the vowel sounds of English, hence the extensive use of less familiar symbols for the vowels. Pronunciations in the text are in Southern British English and are given between slashes; for example, the pronunciation of the word *English* would appear as /ˈɪŋglɪʃ/. (The symbol ˈ indicates primary stress.)

CHAPTER ONE

Introduction

One of the discomfiting things about living abroad is that people tend to treat you as an official representative of your home nation, taking you to be personally responsible for any aspect of it that they disapprove of. This does not stop at your government's current policies, with which, just conceivably, you might be vaguely associated, but extends to regrettable incidents in your nation's history and to aspects of your national customs and culture.

Thus it was that I found myself, some years ago, defending the English language over lunch in a staff canteen in Paris. I carried the flag as best I could but was hopelessly outnumbered, and the discussion was wound up by a middle-aged French lady, to general approval, with the statement, 'There are no rules in English, only exceptions.' Thinking about this later, I decided that she was probably talking about the spelling, and I also decided that there was some truth in what she said.

I became more seriously interested in spelling when I returned to England in 1977 after working for some time in Africa. The Adult Literacy Campaign was under way at that time and there was a possibility that I might do some research connected with it. In the event, these plans came to nothing, but I discovered, in the preparatory conversations I had with adult literacy specialists, that many of the people who were enrolling for tuition wanted help with spelling rather than reading or handwriting. This was later confirmed by a large survey of adult literacy students (Gorman 1981) which reported that sixty-seven per cent had much more difficulty in writing than in reading and that seventy-two per cent

had particular difficulty with spelling.

In Lesotho, the country I had just returned from, the spelling of the language was straightforward; if you could pronounce a word, you could spell it. This is true of many African languages, especially those written down for the first time in this century, and it is true also, in varying degrees, of some other languages written down for much longer, such as Spanish and Italian. There is no linguistic reason why the spelling of English has to be more difficult; the difficulty is just a product of its history. It seemed absurd to me that people should be enrolling for adult literacy tuition simply in order to spell the words of their own language. There is, perhaps, some unavoidable difficulty in expressing one's thoughts in writing, but there need not be any in just spelling the words. If people are having trouble with English spelling, it's the spelling that needs attention, not the people.

My first thought was that the orthography – the way that English is spelt – should be made easier, but I soon discovered that I was not the first person to think of this. People have been proposing schemes of spelling reform for at least four centuries, the movement reaching its apogee about ninety years ago when it enjoyed the support of eminent linguists, writers, educators, industrialists, politicians and many more, funded, in America, by a quarter of a million dollars from Andrew Carnegie (Scragg 1974, Venezky 1980). Despite this support and despite the eminent reasonableness of the cause – albeit somewhat overstated by the devotees – the movement has had almost no impact at all on English spelling. I see no reason to expect that it will succeed now when it has failed so completely in the past.

I had reached this melancholy point in my thinking when I became interested in computers, initially through my work in social research rather than because of any connection with spelling. As I learnt more about them, it occurred to me that computers could provide a partial answer to the problem.

It is debatable whether computer technology will ever completely replace printing and handwriting, but it is undeniably making inroads. At this moment, I am sitting at a computer terminal, typing these sentences on a keyboard; they are being stored on a disc, and the sentences I have just typed are displayed on a screen in front of me. Ever larger amounts of written material

are being created or processed like this and, as personal computers become both cheaper and more powerful, even humble documents such as students' essays, committee minutes, personal letters and the like will be routinely produced in this way.

With handwritten, typed, or printed material, the message is locked to the medium. You write a note on a piece of paper, and it is that very piece of paper, with your writing on it, which gets read. If you want to change something you've written, you have to either cross it out, in which case the reader sees the crossings out, or throw away that piece of paper and write it all out again on a new one. In a computer, the storage of text is completely distinct from its presentation. The text is stored as electronic dots on disc or tape or chip. In this electronic form, it is malleable; you can put things in, take things out or move things around. Programs can act on the text. It is very easy, for instance, to turn all the small letters into capitals or to encrypt the text into a secret code. More usefully, programs can check it for typing mistakes, or arrange it into pages for printing, or generate an index from it or translate it (after a fashion) into a foreign language. When it needs to be presented for human consumption, it can be converted into visible (or even audible or tactile) form by whatever devices are available; it may be displayed on a screen, or printed on paper, or transferred to film or fed through a voice box or a braille device.

To anyone who knows about word-processing, I am labouring the obvious, but it's an important point. The reason that poor spellers are unhappy about their spelling is not, generally, that their spelling is so bad that no-one can understand what they write (though the spelling of very poor spellers can be that bad) but rather that poor spelling gives a bad impression; people tend to think that someone who can't spell is dim-witted or slipshod. If the poor speller writes or types straight onto paper, he lays himself open to this since the reader sees exactly what's been written, misspellings and all, but if the text first goes into a computer, the computer may be able to correct the misspellings before the text gets printed or displayed for any readers.

What poor spellers need, and what I have tried to produce as part of my research, is a piece of software capable of doing this job. It should be able to take a text written by a poor speller, detect the misspellings, guess what words the writer intended and suggest

the correct spellings. It should do the job about as well as a good typist would – someone who had no knowledge of the writer or of the content of the document but who knew English and whose spelling was good. If the text was incomprehensible, the computer would not be likely to correct it properly, but neither would the typist.

Spelling, I have discovered, is a topic on which everyone has an opinion. Some critics accuse me of exaggerating the problem. Most people have no great difficulty with spelling, they say, and anyway poor spelling is not a problem so long as people can still understand what you write. They often follow this with references to people who have achieved greatness (Winston Churchill is a favourite) despite this supposed handicap of poor spelling.

My impression, first, is that even good spellers are not as good as they think they are. In tests I have administered informally to various groups of graduates, only a few (under twenty per cent) have been able to spell *minuscule, sacrilegious* and *ecstasy*, and I often encounter misspelt versions of *occurrence, accommodation* and other perfectly ordinary words. At the other end of the scale, a survey in 1981 of a representative sample of people in Great Britain in their early twenties (Hamilton and Stasinopoulos 1987) found that ten per cent had had difficulties with writing or spelling since leaving school, the difficulties being sometimes quite severe:

> I went for a job as an ambulance driver and the writing and spelling let me down. It stops me getting a better job, a more secure one.
>
> It's embarrassing – very embarrassing in so many ways. For instance, if I send the kid to a shop, I can't write out what I need – I can't spell. I'm a nice writer but I can't spell.

Figures for the population in general are hard to come by. However, a survey was carried out in October 1992 in which a thousand people aged over sixteen were asked to spell six common words which were known to have troublesome spellings.[1] They could respond out loud or on paper, as they wished. The results are given in Tables 1.1 and 1.2.

Figures from another source are broadly in line with these results. In a test given to all the fifteen-year-olds in schools in Cambridge U.K. in 1970, it was found that seventy-five per cent couldn't spell *disappoint* and eighty-five per cent couldn't spell

Table 1.1 Per cent of adults spelling each word correctly

Test word	*Correct*	*Incorrect*	*No attempt*	*Total*
height	84%	10%	6%	100%
business	65%	26%	9%	100%
sincerely	61%	26%	13%	100%
necessary	58%	30%	12%	100%
separate	51%	40%	9%	100%
accommodation	27%	62%	11%	100%

Table 1.2 Adults' success in spelling the six words of Table 1.1

Got all six right	17%
Got five right	19%
Got four right	19%
Got three right	14%
Got two right	11%
Got one right	11%
Got none right	9%
Total	100%

embarrass.[2] Having difficulty in spelling ordinary words is not a minority problem.

Even if native speakers of English coped well with the spelling, there would remain the problems of foreign speakers. English is an international language. The great majority of users of English worldwide are people who use it as a second language.

As to the idea that poor spelling doesn't really matter, the quotations above suggest otherwise, as do the enrolments for adult literacy tuition. In the first two years of the Adult Literacy Campaign (1975-77), over 125,000 people enrolled in England and Wales (Gorman 1981). Even allowing for a small group of non-native speakers of English, and for those needing help with reading and handwriting rather than with spelling, this indicates the extent to which poor spellers themselves regard their spelling as a problem.

The fear of making spelling mistakes is shown also by studies of schoolchildren's writing (Moseley 1989). Children avoid using words they cannot spell. For many children, this rules out such a

large proportion of their vocabulary that their writing is rendered dull and repetitive. It is not simply that they avoid rare words. Even high frequency words are danger spots if the spelling is troublesome, causing the children to cast about for safer ways of saying things, or safer things to say.

Other critics, while not accusing me of exaggerating, accuse me of threatening to make an already serious problem even worse. They fear that a computerized spelling corrector, if widely used, would lower (they tend to say 'yet further') the general standard of spelling. If children can wave a wand over their text and have it corrected, they will not apply themselves to remembering the correct spellings.

I honestly don't know how likely this is. It seems plausible, and yet the opposite also seems plausible; if people's attention is drawn (by a computer) to words that they habitually misspell, and if they become accustomed to reading their own text without misspellings, they may tend to remember the correct spellings. People with severe spelling problems, and their teachers, seem to have no doubt about the value of computer spellcheckers (Singleton 1991, Innes 1990); their complaint is that the spellcheckers do not do the job well enough.

However, I must admit that, even if some decline in people's spelling ability were confidently predicted, I don't think this would be a strong enough argument for prohibiting the development or use of the software. There is no intrinsic virtue in memorizing the quirks of English spelling. That the orthography has become fixed in its present form is largely the result of the locking of the message to the medium which is a feature of writing, typing and printing straight onto paper. For several hundred years, people have had to adapt themselves, with varying degrees of failure, to this technology. The technology can now adapt itself to people, and that is surely the right way round.

There is no software at the moment that can correct the text of a poor speller as well as a good typist would. While simple programs can correct a certain proportion of misspellings, further progress requires a more complicated approach, based on knowledge of the misspellings that poor spellers are likely to make and some understanding of why they make them. The structure of the book follows from this.

The first half of the book is about spelling, the second about computers. Chapter Two describes how English spelling came to be in the state that it's in today. In Chapter Three I summarize the debate between those who propose radical change to the system and those who favour keeping it as it is, and I show how computerized correction can be seen as providing at least some of the benefits that have been claimed for spelling reform. Too much of the literature on computerized spellcheckers describes tests based on collections of artificially created errors; Chapter Four looks at the sorts of misspellings that people actually make, to see more clearly the problems that a spellchecker has to face. Chapter Five looks more closely at the errors that people make when they don't know how to spell a word, and Chapter Six at the errors that people make when they know perfectly well how to spell a word but for some reason write or type something else.

Chapter Seven begins the second part of the book with a description of the methods that have been devised over the last thirty years for getting computers to detect and correct spelling errors. Its conclusion is that spellcheckers have some way to go before they can do the job we would like them to do. Chapters Eight to Ten describe a spellchecker that I have designed which attempts to address some of the remaining problems, especially those presented by badly spelt text.

In 1982, when I began this research, there were no spellcheckers that would do anything useful with a sentence such as, 'You shud try to rember all ways to youz a lifejacket when yotting.' That my spellchecker corrects this perfectly (which it does) is less impressive now, I have to admit, than it would have been then, simply because there are now a few spellcheckers on the market which do make a reasonable attempt at errors of that kind. My spellchecker does, however, handle some classes of errors that other spellcheckers do not perform well on, and Chapter Eleven concludes the book with the results of some comparative tests, a few reflections on my spellchecker's shortcomings and some speculations on possible developments.

Notes

1. The survey was carried out by Gallup for the Adult Literacy and Basic Skills Unit (ALBSU). These figures are taken from a press release prepared by ALBSU. The report gives the number of people interviewed as 'just over 1,000'. The questions were put in one of Gallup's regular Omnibus surveys. The respondents formed a quota sample stratified by sex, age, social class and working or non-working, selected by the interviewers at 110 points in England and Wales.
2. The study was conducted by Dr Margaret Peters (1970), but pressure of work prevented her from analysing the data. The material now forms one of several collections of spelling errors available in computer-readable form (Mitton 1985).

CHAPTER TWO

A short history of English spelling

I said in the Introduction that the spelling of English did not have to be more difficult than, say, the spelling of Spanish or Italian, and that the difficulty arose from its history. I hope in this chapter to justify that statement by showing how the orthography changed from earliest times until it became fixed into something very like its present form about three hundred years ago.

As England emerged from the Dark Ages, writing was widely practised by the clergy for both secular and religious purposes. At first it was mostly in Latin, but from the seventh century more of it was in English and, by the time of Alfred the Great (who died in 899), a number of English books, mainly translations from Latin, were being produced. There was no fixed orthography for English; a scribe's spelling of a particular word depended partly on the local conventions, partly on his dialect and partly on choice – he might spell the same word in different ways in the same manuscript. The idea that there is one and only one 'correct' spelling for a given word is relatively modern (Hogg 1992).

The middle of the tenth century, however, saw a vigorous monastic revival led by Æthelwold, Bishop of Winchester, and, with it, the widespread adoption of a standard form of written English, based on West Saxon. The following example shows the first line of the Lord's Prayer translated by Ælfric, one of the leading writers of the period, in about 990:

þu ure fæder þe eart on heofonum sy þin nama ȝehalȝod.

Obviously some changes are made in presenting a line of manuscript in print. The main difference is that the words are

clearly separated by spaces in the printed form but not so clearly, or not at all, in the original.

The language, of course, is Old English, not readily intelligible to a speaker of modern English, but the words are recognizable as ancestors of present-day words – *fæder-father, heofonum-heaven, nama-name.* Nearly all the letters – the ones that had been taken from the Roman alphabet – are the same as those in use today. The unfamiliar ones in this example are æ, called *ash* and pronounced like the *a* of *cat,* þ, called *thorn* and pronounced like the *th* of *this* or *thistle,* and ȝ, called *yogh*. This last one has various pronunciations; the one at the beginning of ȝehalȝod is like the *y* of *yellow* while the one in the middle corresponds to a sound we do not have any more, a voiced version of the Scottish *ch* in *loch.* The following approximation to the pronunciation is given by Scragg (1974):

θu: u:rə 'fædər θe ært ɑ̃n 'hevənən 'si: θi:n 'nɑ̃mə je'hɑ:lɣəd

Readers not familiar with the phonetic alphabet can get a (very) rough idea of how it might have sounded by reading this out loud:

thoo oorer fadder, the art ong hevernern, see theen nongmer yeharlgerd.

(This is my own rendition. I imagine it would make an Old English scholar cringe. The *a* of *art* should be pronounced like the *a* of *cat.* The *ong* is meant to be like the vowel sound of the French word *blanc.* There is stress on the first syllable of *hevernern* and on the second of *yeharlgerd.*)

The adoption of the Roman alphabet, with only minor extensions, was to have a lasting effect on the orthography, for the Roman alphabet, of course, was devised for representing a different language. Any linguist devising an alphabet for English from scratch would devise one with more letters, especially for vowel sounds, but we are stuck with the Roman ones. This partly accounts for the number of letter-pairs in modern English spelling, like *th, sh, ea, ou* and so on, where the pair is quite different from the two letters sounded separately.

So far as scholars can tell, the relationship of the spelling to the pronunciation was more straightforward in Old English than at any later time; there were hardly any 'silent' consonant letters, for instance, to catch the unwary scribe. At the end of the tenth century, there was a single system, with only minor variations,

throughout England. In terms of simplicity, this was the high point of English spelling.

The Norman invasion reduced the amount of writing in English, and the stable spelling system began to fall apart. Manuscripts in English began to show signs of regional dialect and local spelling conventions and the influence of French and Latin. The spoken language was also changing; a sound like the *ch* of the Scottish *loch,* for example, gradually disappeared in the middle ages from words like *night* and there was a good deal of variability in its spelling (Milroy 1992).

The invasion also brought a host of Norman-French words and spelling patterns, and this influence was prolonged by the rise of metropolitan French as the courtly language of international diplomacy in the middle ages; perhaps as much as forty per cent of today's English vocabulary is derived from French. The words that arrived in the middle ages, especially during the Anglo-Norman period, are now completely assimilated into English and have no foreign tinge at all, such as *royal, gentle, chance* and *danger* (as contrasted with those that entered the language much later, in the sixteenth to eighteenth centuries, such as *cordon, vogue, moustache, clique, salon* and so on). Even within the middle ages, there was a difference between those that arrived early, which came from Norman French, and those that arrived later, from Parisian French. Occasionally the same word was borrowed twice, once from Norman French and later from Parisian French: *warden, cattle* and *gaol* are from Norman French, *guardian, chattel* and *jail* from Parisian (Strang 1970).

Many of the scribes would be bilingual and much of what they were copying was in French, so French spelling patterns tended to creep in when they were writing English. Sometimes a French spelling pattern simply replaced a perfectly good Old English one; *qu,* for example, replaced the earlier *cw* in words like *queen* and *quick.* At other times a French pattern provided a solution for an English orthographic problem. For example, it had not been felt necessary in Old English to have a special letter for the sound /tʃ/ (the initial sound of *chin*) but, by 1200, pronunciation had changed and writers needed some way of marking it; they solved the problem by taking *ch* from French. (The French *ch* was pronounced at this time like the *ch* of *chin;* only later did it come to be

pronounced /ʃ/ like the *sh* in *shin,* as it is today.)

Often, however, the new French spellings just caused confusion. A good example of this is the use of initial *h.* French scribes took to writing French words with *h* if they were obviously derived from Latin words beginning with *h,* even though the *h* was not pronounced in French – modern French *habile, honneur, hôpital* and so on. That the French scribes favoured the initial *h* in writing was not entirely due to their fondness for Latin; the amount they got paid for a legal document depended on the number of letters in it, so superfluous letters may have been a source of income (Ewert 1933).

When these words were introduced into English, the initial *h* was retained in the spelling, though manuscripts show a lot of variation in their use of it (Milroy 1992). With some of these words, like *able*, the *h* never caught on; some of them, like *honour*, retained the *h* in the spelling though it continued to be silent, as in French, and some of them, like *hospital,* have had their pronunciation changed so that the *h* is now sounded. This process has continued down the years; the words *humour, hospital* and *herb* were not uncommonly pronounced without the *h* as recently as the early 1900s. (*Herb* is still pronounced without an aitch in America.) Oddly, though dropping one's aitches has been widely regarded as uncouth since the eighteenth century, the upper classes pronounce some h-words without the *h,* favouring an old-fashioned pronunciation in this respect as in certain others (Strang 1970) – it can be rather posh to say *an hotel* (with no *h).* Perhaps this explains why people in formal settings, such as conferences of computer scientists, sometimes make a point of using the *an* form of the indefinite article (because it seems somehow more refined) before words such as *hierarchy* or *historical,* though they are equally careful to aspirate the *h* (not wanting anyone to think they drop their aitches).

Many people today feel that it is contrary to common sense for the spelling of a word to be different from what the pronunciation would lead you to expect. Yet, even from medieval times, people have been respelling words in a way that made the spelling more remote from the pronunciation. The silent *h* is one example; another is the replacement of *u* by *o* in words like *come, love, monk* and *wonder*. The letters *u, m* and *n* were written as sequences of

short, unligatured downstrokes, known as 'minims' (Lass 1992). Present day *v* was written as a *u* and *w* as *uu*, as its name implies. To further complicate matters, *i* was written as a single, undotted minim. So a run of four minims could represent *w, un, uv* etc; a run of five could be *um, mu, wi,* and so on. To help disambiguate such runs, scribes borrowed the late Latin practice (Scragg 1974) of writing *o* for *u* (Venezky 1976). They also sometimes used *y* for *i* for the same reason (Strang 1970).

Though diversity was the main feature of English spelling in the middle ages, a standard form eventually emerged. Whereas Winchester had been the earlier centre of political power and the dialect of that area had been the basis of the standard orthography, power shifted to London from the eleventh century and, in time, documents coming from London formed the basis of a new standard. These were writings issuing from the Chancery – legal documents rather than religious or literary ones. For many years these were in Latin or French but, from about 1430, they began to be written in English. These were important documents, circulated round the country, and the spelling was used as a model by scriveners – professional writers of secular manuscripts (records of guilds and boroughs and the like).

The spelling naturally reflected the speech of the London area, which had been influenced mainly by the dialect of the east midlands since many of London's inhabitants of the time had come from there, but it was not a phonetic system; it was, like today's orthography, a written form of the language existing alongside the spoken one. The Chancery scribes did not invent a new spelling system; they settled on a restricted set of the spellings that were current at the time. In many cases the ones they had adopted by 1450 are the ones we have today, such as *such* in preference to *sich, sych, seche* and *swiche,* and *which* in preference to *wich,* so the spelling of these documents seems reasonably familiar to a reader of modern English (Blake 1992). Here is a small sample:[1]

> The kyng by þadvise and assent of the lordes spirituell and temporell beyng in this present parlement woll and grantith þat þe said Sir Iohn Talbot haue and occupie the saide office of Chaunceller of Irelond by hym self or by his sufficient depute there after the fourme of the kynges lettres patentes to hym made þerof. the which lettres patentes ben thought gode and effectuell and to be approved after the tenure of the same Also þat þe grete seal of þe saide

> lond belongyng to þe saide office. which þe said Thomas hath geton vn to hym be delyuered to þe said Sir Iohn Talbot or his sufficiante depute hauyng power of hym to resceiue hit.

The next great influence on English spelling was the revival of interest in Greek and Latin literature which was in full spate by the early sixteenth century. Latin was known, and revered, by most educated people, and they felt that, since so many English words were derived ultimately from Latin (this was because so many of these words had come into the language via French), the spelling of the words should show this. They therefore created new spellings to make the words look more like their Latin originals. Table 2.1 presents a few examples (Scragg 1974):

Table 2.1 Some etymological spellings

Old	*New*	*Old*	*New*
assoil	absolve	colere	choler
amonest	admonish	dette	debt
cors	corpse	doute	doubt
descryve	describe	receite	receipt
langage	language	samon	salmon
peynture	picture	ceptre	sceptre
trone	throne	vitailes	victuals

That so many of the new spellings now seem the more natural is because of a process known as 'spelling-pronunciation'; in the intervening centuries, people have changed their pronunciation of the words to bring it into line with the spelling. This has clearly happened with those on the left in Table 2.1 but not with those on the right.

The process has continued to the present day. Table 2.2 shows a few words where a spelling-pronunciation is currently competing with an older one, or has recently displaced it (Strang 1970). This process can also affect the pronunciation of proper names. For example, many English place-names end in *ham*, an old word for 'village' or 'manor'. Where the first part of the name happens to end in *s* or *t*, the resulting *sh* or *th* looks like a familiar orthographic unit and it is often pronounced accordingly, as in *Amersham, Evesham, Grantham* and *Waltham* (Clark 1992).

Table 2.2 Some recent spelling-pronunciations

Spelling	*Old pronunciation*	*New pronunciation*
again	/əˈgen/ *(a-genn)*	/əˈgeɪn/ *(a-gain)*
conduit	/ˈkʌndɪt/ *(kundit)*	/ˈkɒndwɪt/ *(kondwit)*
forehead	/ˈfɒrɪd/ *(forrid)*	/ˈfɔːhed/ *(forhedd)*
nephew	/ˈnevjuː/ *(nev-yoo)*	/ˈnefjuː/ *(neff-yoo)*
often	/ˈɒfn/ *(off'n)*	/ˈɒftən/ *(offt'n)*
waistcoat	/ˈweskɪt/ *(weskit)*	/ˈweɪskəʊt/ *(waisscoat)*

The new spellings which the Latin scholars invented were sometimes based on faulty etymology. They changed *sisoures* and *sithe* to *scissors* and *scythe* on the false assumption that these words were derived from Latin *scindere* (to cut). They put an *s* into *island* (medieval *yland*) by analogy with *isle* (from French) and *insula* (Latin) though the word came from Old English and had never had an *s.* They put a *d* in *advantage*, an *l* in *emerald* and an *h* in *anchor* which had no business being there (Scragg 1974). Etymologically justified or not, however, a large number of these quirky spellings remained.

These new spellings did not always get established. *Saint*, for example, was respelt *sanct,* but then *saint* reasserted itself (Scragg 1974); *conceit* and *deceit* were written *conceipt* and *deceipt* for a while (Strang 1970). It is possible that more of them would have gone the way of *sanct* had it not been that the orthography was becoming fixed at this time, due to the rise of printing, though the initial effect of the new technology, introduced by Caxton in the 1470s, was not to stabilize spelling but to confuse it still further.

One effect was to force out of use the handful of non-Latin letters that had survived from the Old English alphabet. The founts of type were manufactured on the continent and did not contain these letters. Thorn (þ) was still in use in the fifteenth century, though it had changed in shape and had come to look like a *y*, sometimes backwards, so printers used the nearest-looking letter they had – a *y.* It was especially popular in abbreviations of *the* and *that* – y^e and y^t. It survives tenuously to this day in mock-archaic signs such as *Ye Olde Tea Shoppe* where the first word is not really *ye* but an old spelling of *the.* Yogh (ȝ) was also still in use. In England it was replaced either by *y* or by *gh* (hence its name),

depending on its role in the word. In Scotland, printers used a *z*, thus creating spellings that have survived in certain names; the middle letter of *Dalziel* and *Menzies,* for example, is really a yogh, not a *z*, and there is no /z/ in the Scottish pronunciation – /di:'jel/ and /'mɪŋɪs/, something like *Dee-yell* and *Ming-iss* (Lass 1992).

Most of the early compositors were foreigners and it took some time before the printing houses developed their house styles. Caxton himself had spent much of his life in the Low Countries and this was reflected in his spelling – he is responsible, for instance, for respelling *gost* as *ghost*. Different compositors used different spellings, and any one compositor would use different spellings of the same word in different places; a pamphlet of 1591 about catching rabbits spells *coney* (an old word for 'rabbit') as *cony, conny, conye, conie, connie, coni, cuny, cunny* and *cunnie* (Baugh and Cable 1978). One reason for this was to adjust the lengths of lines so as to get a straight right-hand margin. Of the two principal compositors who set Shakespeare's First Folio (published in 1623), one favoured *doe, goe* and *here,* while the other favoured *do, go* and *heere,* but each occasionally borrowed the variant preferred by the other if he needed to lengthen or shorten a line (Scragg 1974).

While it might have suited the printers to use alternative forms, it did not please their readers. From about the middle of the fifteenth century, people began to voice criticism of the state of the orthography and in particular the practices of printers. An early spelling reformer called John Hart, writing in 1551, argued that 'vicious' writing 'bringeth confusion and uncertainte in the reading' (Salmon forthcoming). He criticized, among other things, the use of superfluous letters, as in *stoppe* (where *stop* would do), and additions made by the etymologists such as the *b* in *doubt,* the *g* in *eight,* the *l* in *souldiours* and the *o* in *people.*

Though Hart's proposals for reforming the orthography along phonetic lines were not widely supported, many shared his concern. Chief among these was an eminent headmaster called Richard Mulcaster who published in 1582 an influential book on the teaching of reading and writing. He rejected wholesale reform as unnecessary and impractical but encouraged the adoption of certain of the variant spellings then in use in preference to others and devoted the last fifty-five pages of his book to an alphabetical list of recommended spellings. Though not all of his preferred

spellings came eventually to be adopted, his principles and his list were taken up by another schoolmaster called Edmond Coote who published *The English Schoole-maister* in 1596. While Mulcaster's book was a learned work for the instruction of teachers, Coote's was more like a school textbook. It evidently filled a need – it ran to over forty editions and was still being published in the early eighteenth century (Scragg 1974).

By 1600 there was a trend to uniformity in the spelling of printed matter and by 1650 it was largely complete. The reasons seem to have been the desire of schoolteachers, on the one hand, for an accepted orthography which they could teach to their pupils – several more spelling books followed Coote's – and the desire of printers, on the other, to meet the expectations of their readers. The idea of 'correct' spelling, coupled with the technology of print, was self-reinforcing. Simply because many copies of a given work could exist, many people could share the same standard. The spellings of the established printing houses were codified and taught to schoolchildren; the children would grow up to expect the same spellings in their reading matter, and the printers would be careful to stick to them. There was no place for innovation in this cycle, and the argument for conservatism grew stronger as the mountain of material in print grew ever larger.

The following extracts illustrate the change from an orthography which, though readable, clearly belongs to earlier times, to one which is hardly different from that used today:[2]

Roger Ascham, 'Toxophilus', 1545

And as for ye Latin or greke tonge, every thing is so excellently done in them, that none can do better: In the Englysh tonge contrary, every thinge in a maner so meanly, bothe for the matter and handelynge, that no man can do worse. For therein the least learned for the moste part, have ben alwayes moost redye to wryte. And they whiche had leaste hope in latin, have bene moste boulde in englyshe: when surelye every man that is moste ready to taulke, is not moost able to wryte. He that wyll wryte well in any tongue, muste folowe thys councel of Aristotle, to speake as the common people do, to thinke as wise men do; and so shoulde every man understande hym, and the judgement of wyse men alowe hym. Many English writers have not done so, but usinge straunge wordes as latin, french and Italian, do make all thinges darke and harde . . .

Richard Mulcaster, 'Elementarie', 1582

It were a thing verie praiseworthie in my opinion, and no lesse profitable then praise worthie, if som one well learned and as laborious a man, wold gather all the words which we vse in our English tung, whether naturall or incorporate, out of all professions, as well learned as not, into one dictionarie, and besides the right writing, which is incident to the Alphabete, wold open vnto vs therein, both their naturall force, and their proper vse: that by his honest trauell we might be as able to iudge of our own tung, which we haue by rote, as we ar of others, which we learn by rule. The want whereof, is the onelie cause why, that verie manie men, being excellentlie well learned in foren speche, can hardlie discern what theie haue at home . . .

John Chamberlain, from a letter describing Sir Walter Raleigh's execution, 1618

When the hangman asked him forgivenes he desired to see the axe, and feeling the edge he saide that yt was a fayre sharpe medicine to cure him of all his diseases and miseries. When he was laide downe some found fault that his face was west-ward, and wold have him turned, whereupon rising he saide yt was no great matter which way a mans head stoode so his heart lay right. He had geven order to the executioner that after some short meditation when he strecht forth his handes he shold dispatch him. After once or twise putting foorth his handes, the fellow out of timerousnes (or what other cause) forbearing, he was faine to bid him strike, and so at two blowes he tooke of his head, though he stirred not a whit after the first. The people were much affected at the sight insomuch that one was heard say that we had not such another head to cut of.

Edward Phillips, 'The New World of English Words', 1658

Whether this innovation of words deprave, or inrich our English tongue is a consideration that admits of various censures, according to the different fancies of men. Certainly as by an invasion of strangers, many of the old inhabitants must needs be either slain, or forced to fly the Land; so it happens in the introducing of strange words, the old ones in whose room they come must needs in time be forgotten, and grow obsolete . . .

Though there are advantages in having a stable orthography, it is unfortunate that the fixing took place when it did – roughly between 1550 and 1650. Another hundred years of confusion might have allowed some of the more awkward Latinate respellings to disappear (*debt, receipt* and the like) and it would also have allowed spellings to adjust to modern pronunciation. The period 1400 to 1600 was a period of unusually rapid change in the pronunciation of English, referred to by scholars as the *Great Vowel Shift* (Wrenn 1949). To take just one example, the words *meat* and

meet used to be pronounced differently, and their spellings were a reasonable representation of their pronunciation. During this period, however, both vowels changed and also fell together to their present pronunciation, but the spellings remained fixed in their previous forms. There used to be a word *quean,* meaning *harlot,* but, by the same process, it came to be pronounced in the same way as *queen* and fell out of use (Samuels 1972). Consonants also were affected in this period of change. The spelling of *knight,* for instance, was a fair representation of how it had been pronounced in 1400, but, by 1600, the sounds to which the *k* and the *gh* corresponded had disappeared from this word. The spelling, however, had become fixed by then.

Though the spelling of published books was stabilized in the seventeenth century, the spelling of private, handwritten documents remained variable for much longer. In Elizabethan times, people did not have the feeling, as they have today, that there was just one correct spelling of any given word; it was acceptable even for an educated writer such as Queen Elizabeth herself, writing in the 1580s, to spell a word in two different ways in the same letter. But as the idea of 'correct' spelling gained force, people felt they should try to conform to book-spelling in their own writing. Coote's spelling book of 1596 was dedicated to people who, for want of the ability to spell, 'are ashamed to write unto their best friends.'

As spelling grew in importance as a subject in elementary schools, it became possible to judge the level of someone's education, in a rough and ready way, by looking at their spelling, whence the long history of mockery which poor spellers have had to endure. Women suffered particularly from this; they generally received little education so their spelling tended to be idiosyncratic and was frequently the butt of supercilious mirth from the more educationally privileged men. But it was not confined to women; Lord Chesterfield, writing to his son in 1750, asserted severely that correct spelling 'is so absolutely necessary for a gentleman, that one false spelling may fix a ridicule upon him for the rest of his life.' A book about letter-writing of 1800 bluntly states that 'ignorance [of spelling] is always considered a mark of ill-breeding, defective education, or natural stupidity.'

Good spelling also became a necessary qualification for many types of employment; a schools inspector of the nineteenth century observed that 'out of 1,972 failures in the Civil Service examinations, 1,866 candidates were plucked for spelling; that is, eighteen out of every nineteen who failed, failed in spelling.' Schools responded to this pressure with large amounts of spelling drill; an educational researcher visiting a class of ten-year-olds in America in the late nineteenth century found them being drilled on the words *exogen, cylindrical, coniferal, resinous* and *whorls,* when results from his own spelling test found that many of the same children were having trouble with words such as *running, slipped* and *believe* (Venezky 1980).

Since schools are supposed to teach spelling, the general standard of people's spelling has come to be regarded as a measure of how well or badly the schools are doing their job. Those who have been alarmed by what they see as a decline in British educational standards since the war have often cited people's spelling as a clear indicator: 'an external examiner of colleges of education writes that it is common to find many students who write *his* for *is,* who do not know the difference between *their* and *there* or *where* and *were,* who cannot punctuate and cannot spell.'[3] More recently, Prince Charles's derogatory remarks about his secretaries' spelling received wide publicity, and the Secretary of State for Education instructed that candidates in public examinations should be penalized for poor spelling regardless of the subject of the examination.

It is significant that I have reached this stage in my account of the history of English spelling without using the word *dictionary.* Contrary to the popular view, dictionary makers do not decide how words are to be spelt. They merely record current practice. Even the enormously authoritative Dr Johnson, whose dictionary was published in 1755, only set down the already established spellings of words; he himself lamented the state that the orthography had got itself into, pointing out the inconsistency of *convey* and *inveigh, deceit* and *receipt* and so on, but felt that it would be hopeless to attempt reform, concluding in his preface that 'to change all would be too much, and to change one is nothing' (Johnson 1755).

Dictionaries do, however, tend to fix the spellings of words. If people come to think that there are correct and incorrect spellings, they want a reference book that will tell them the correct ones, and this is partly what dictionaries do. Bilingual dictionaries – English-Latin, English-French – had existed since the fifteenth century, but the idea of a book that explained English words in English for native speakers of English did not arise till the late sixteenth. At first these were spelling lists, perhaps with some notes on meanings, concentrating on harder words. It was not until the eighteenth century that people produced dictionaries that attempted to cover all the words of English, or anyway a very large number of them. Dr Johnson's dictionary was perhaps the final nail that pinned the orthography down; everyone could now use the same authoritative work of reference, whether for private writing or for publication. The *Oxford* and the other major dictionaries play that role nowadays, and their authority is unchallenged. It does not matter how convinced you are of the spelling *extacy*, or how often you think you've seen it written that way, or what reasons you can give for why it should be spelt like that; if the dictionary says *ecstasy*, you're wrong.

Today's writers do not use exactly the spellings of Johnson's dictionary – he has *horrour, terrour, musick* and *physick*, for example – but the changes since Johnson's time have been minor. The only important part of the story that remains is the curious tale of Noah Webster, whose dictionary had the same status in America that Dr Johnson's had in Britain. He made a fortune from a conservative spelling book first published soon after independence, and then ploughed much of it into schemes of spelling reform, largely ineffective. In the dictionary that he then wrote he incorporated a few of the reformed spellings that he favoured; it is not that he invented these, but rather that he chose to include certain variant spellings then in use in preference to others. Not all of the reformed spellings in the first edition of his dictionary caught on – he included *ake, crum, fether, ile* and *spunge* (Venezky 1980) – but most of the now-familiar distinctively American spellings are Webster's – *theater, meter* for *theatre, metre; honor, favor* for *honour, favour; defense* for *defence, check* for *cheque* and *traveling* for *travelling* (Baugh and Cable 1978). It seems that he promoted his reformed spellings with some vigour; a biography of Webster contains the

following anecdote:[4]

> The present printer [1881] of Webster's Dictionary remembers that when he was a boy of thirteen, working at the case in Burlington, Vermont, a little pale-faced man came into the office and handed him a printed slip, saying, 'My lad, when you use these words, please oblige me by spelling them as here: *theater, center,* etc.' It was Noah Webster traveling about among the printing-offices, and persuading people to spell as he did.

Some American spellings have infiltrated British English, often with a special meaning – *computer programs* as opposed to *television programmes,* for instance – but nationalism seems to have kept most of them out.

For over a thousand years, since the establishment of Old English orthography, people have been changing and extending English spelling in various ways, some of which seem to us nowadays rather odd; they have never demolished the structure to start again from scratch, but have rather modified and extended what was there already. The monks of Anglo-Saxon England, the Norman invaders, medieval French scribes, Latin scholars of the Renaissance, Elizabethan schoolmasters and the printers and lexicographers of the Enlightenment have all left their mark on the orthography. Add to that the great capacity of English for assimilating words, and spellings, from other languages – *spaghetti, moccasin, jodhpurs, sheikh* – and the result is a jumble of systems, subsystems and exceptions. It didn't have to be like that, but that is how it is, and, despite the strenuous efforts of spelling reformers, it looks set to stay that way.

Notes

1. This sample is taken from an anthology (Fisher et al. 1984) and is quoted by Blake (1992).
2. All these extracts are taken from Baugh and Cable (1978).
3. This and the quotations in the preceding two paragraphs are all taken from Scragg (1974).
4. The anecdote is taken from a biography of Webster published in 1882 (Scudder 1882). It is quoted by Baugh and Cable (1978).

CHAPTER THREE

Pros and cons of English spelling

Opinions about English orthography vary over the widest possible range. Some see it as a burden to schoolchildren and their teachers (Pitman 1969), a block on the path to literacy (Dewey 1971), an obstacle to foreign learners of English (Mont Follick 1965) and a persistent nuisance to writers, typists and printers. At the other extreme, no less an authority than Noam Chomsky (Chomsky and Halle 1968) has described it as 'near optimal'. My own position is somewhere in between. It seems to me implausible that a system so knocked about by history should turn out to be nearly optimal; indeed, such an outcome would be little short of miraculous. The system is full of quirks, and it would make life easier for many people if these were ironed out. On the other hand, I have doubts about the root-and-branch reforms proposed by many reformers; for all its faults, the system has some virtues.

English orthography is basically alphabetic; there is sufficient consistency between letter and sound for it to be possible at least to begin the teaching of reading by sounding the letters individually and showing how they form words. The *b* of *bat* corresponds to /b/, the *a* to /æ/ and the *t* to /t/, so *bat* spells /bæt/. (Items between slashes represent pronunciations in Southern British English – see the table on page x.) The extent to which teachers actually use this method varies a good deal from one teacher to another, but the method is at least possible.

What the critics complain about is that it is not *consistently* alphabetic. The teacher cannot go on to explain *cough* in the same way; *g* is /g/ and *h* is /h/, so how can *cough* spell /kɒf/? No sooner have children learnt the letters of the alphabet and grasped

the idea of letter-sound correspondence than they begin to encounter a host of words where the system breaks down. In fact they begin to encounter them when they have hardly got started since some of the most peculiarly spelt words in the language are among the commonest; the *f* in the word *of,* for example, does not correspond to /f/.

The alphabetic principle is simple. Out of the several hundred distinguishable sounds that the human vocal equipment can produce, any given language uses only a few, though different languages use different sets. English speakers can make coughing noises with the throat and clicking noises with the tongue, but these are not part of the English language, though they may be used in other languages. The smallest units of speech that distinguish one word from another in a language are called the phonemes of the language. Estimates of the number of phonemes in English vary from one authority to another, but, according to one writer (Gleason 1961), there are thirty-six vowels and consonants in English. To devise an orthography for English on the alphabetic principle, you would have an alphabet of thirty-six letters, one for each consonant or vowel phoneme.[1]

One person's rendering of a particular phoneme may differ from another person's, and even the same person's rendering may vary in subtle ways from one occasion to another. For example, the way you pronounce the *p* in *pill* differs perceptibly from the way you pronounce it in *spill.* (Hold a slip of paper close to your lips and you'll see that there's an explosion of air with *pill* that there isn't with *spill.*) But this difference is of no significance in English, so we are not concerned about it and indeed we hardly notice it. So far as the English language is concerned, they are different realizations of the same phoneme. If a phonetician were noting down the sounds of the language, he might want to distinguish between the two, but ordinary users of English would not. Strictly speaking, then, it is phonemes, rather than sounds, that are represented in an alphabetic orthography.

Obviously, English orthography departs seriously from the alphabetic principle. To start with, there are only twenty-six letters to represent these thirty-six phonemes, so letter-pairs are often used instead of single letters to represent a single phoneme, such as *sh, th, oa, oo* and *oe,* occasionally giving rise to momentary

confusion when the letters are *not* to be taken as a pair *(a bishop's mishap, a swarthy warthog, an inchoate moan, a coopers' cooperative, an orthoepist's toecap).* (The dieresis exists for alerting the reader to some of these problems, as in *naïve,* but it is not often used – Fowler (1968) describes it as 'an obsolescent symbol'. A hyphen is sometimes used instead, as in *co-operative* for *coöperative;* more often there is no mark at all – *cooperative, naive.*)

This would be only a minor nuisance if the letters and the phonemes corresponded consistently, but they don't. Many phonemes have several different representations; /f/, for instance, can be written as *f (often), ff (off), ph (graph), gh (enough)* and, in some foreign names, *v (Chekhov).* Conversely, many letters or letter-patterns have more than one pronunciation; consider *ou* in *out, four, through, tough, cough, dough* and *borough.* At times the principle breaks down so badly that one hesitates to say what corresponds with what. To what does the *gh* correspond in *night* and *bought*? Which letters represent the /tʃ/ in *picture* or the /w/ in *choir*? If it's the *sh* in *fashion* that corresponds to the /ʃ/, presumably it's the *ss* in *mission* and the *t* in *nation*? Or is it the *ti*? In which case, was it the *shi* in *fashion*?[2] Viewed purely as an alphabetic system, English spelling is a mess, a fact which spelling reformers enjoy parading at great length.

If English spelling were redesigned on strict alphabetic lines, there would be more letters in the alphabet; each phoneme would be represented by just one letter, and each letter would represent just one phoneme. The benefits are obvious. Children would learn their reading and writing more quickly and more easily, and more of them would reach an acceptable level of competence. Foreign learners of English (who far outnumber native speakers) would find the spelling a reliable guide to pronunciation, and vice-versa. Spelling problems would largely disappear – no more worrying over *occurrence, accommodation, principal, ecstasy* and the like. There would also be a modest reduction in the average number of letters per word, leading to savings in paper, ink and computer storage space and in the time spent actually writing or typing.

People have been proposing schemes of spelling reform for over four hundred years. As I described in the last chapter, many writers in the sixteenth century expressed concern about the variable spelling in printed works. According to the more

conservative view, the important thing was to settle on one spelling for each word and have everyone stick to it, accepting inconsistencies in the system for the sake of stability. But there was another more radical view, namely that spelling should be brought back into line with pronunciation. The proponents of the more radical view were characteristic of the sorts of people who supported spelling reform over the next four centuries – eminent scholars, phoneticians and schoolteachers. They were often as concerned to spread good pronunciation as good spelling, the idea being that books printed in an alphabetic orthography based on the 'best' pronunciation would be a model of how English should be spoken for people who spoke it 'badly' – rural folk, the lower classes, speakers of regional dialects and the like.

Sadly for subsequent generations of schoolchildren, it was the more conservative view that prevailed. As the orthography settled down into its present form in the seventeenth century, even fairly timid attempts at reform were sternly rebuked. Dr Johnson probably summed up the general view on the matter by the middle of the eighteenth century in his condescending reference to spelling reformers as 'ingenious men' who 'endeavoured to deserve well of their country, by writing *honor* and *labor* for *honour* and *labour, red* for *read,* in the preter-tense, *sais* for *says, repete* for *repeat, explane* for *explain,* or *declame* for *declaim.* Of these it may be said, that as they have done no good, they have done little harm; both because they have innovated little, and because few have followed them.'[3]

Interest in spelling reform was rekindled in the late eighteenth century, Benjamin Franklin and Noah Webster taking up the cause in America, and later Isaac Pitman (of shorthand fame) in this country. Their main concerns were the difficulties that children were having in learning to read and the low levels of literacy which they were achieving by the time they left school, a concern that was intensified by the advent of universal primary education in 1870. It seemed obvious to the reformers that the antiquated orthography was the main cause of these difficulties and that a reform on alphabetic lines was the solution. By the turn of the century there were societies for spelling reform on both sides of the Atlantic; they were well funded and enjoyed the support of many eminent and influential people.

The movement came as close to success as it has ever come when President Theodore Roosevelt ordered the Government Printer to use a list of three hundred reformed spellings advocated by the Simplified Spelling Board, an organization funded by the millionaire Andrew Carnegie. The reforms were modest. Some of the three hundred were Webster's preferred forms already in widespread use in America – *honor, theater* and the like; others involved only the deletion of an *e (acknowledgment, judgment* and so on) or the dropping of *ue (catalog, demagog),* though some others were more radical (*pur, dript, husht).* But the order, issued in August 1906, was revoked in December in the face of public outcry. Apart from consolidating the use of those forms which were already accepted as American variants, this small and short-lived victory for the reformers had little effect except to galvanize the forces of conservatism. The movement did not die, but it lost its impetus (Venezky 1980).

Though the alphabetic principle itself is simple enough, there are many decisions to be made in applying it. Consequently, though reformers have agreed on what is wrong with the traditional orthography, they have differed widely in their proposals for a new one. Do you make do with the Roman alphabet or do you supplement it with new characters? The latter is essential if you wish to have an orthography with one letter for each phoneme but it obviously runs into severe practical problems. Or perhaps you go for a half-way solution using diacritics? Do you compromise on the alphabetic principle in order to retain as much continuity as possible with traditional orthography or, on the contrary, do you advocate a complete break with the old system? Whose interests do you have uppermost in your mind in choosing this or that new spelling – English-speaking children learning to read, foreigners learning English, or skilled readers and writers of English who do a great deal of reading and writing? These and other questions have exercised the minds of reformers down the years.

The principal British organization for spelling reform is the Simplified Spelling Society. Their system, devised early this century and modified periodically, accepts that the adoption of new letters would be impractical, but retains a consistent spelling – sometimes with one letter, sometimes with two – for each phoneme, though trying to do this with as little disturbance as

possible to traditional spellings. Here are two samples, the first a sentence already familiar to readers, the second probably unfamiliar (MacCarthy 1969a):

> Forskor and seven yeerz agoe our faadherz braut forth on dhis kontinent a nue naeshon, konseevd in liberti, and dedikaeted to dhe propozishon dhat aul men ar kreeaeted eekwal.
>
> We instinktivli shrink from eni chaenj in whot iz familiar; and whot kan be mor familiar dhan dhe form ov wurdz dhat we hav seen and riten mor tiemz dhan we kan posibli estimaet? . . . At dhe furst glaans a pasej in eni reformd speling looks "kweer" or "ugli." Dhis objekshon iz aulwaez dhe furst to be maed; it iz purfektli natueral; it iz dhe hardest to remuuv. Indeed, its efekt iz not weekend until dhe nue speling iz noe longger nue, until it haz been seen ofen enuf to be familiar.

This is the version of New Spelling devised by Walter Ripman and William Archer in the early years of the century (Ripman and Archer 1940). The Society has made small modifications since then.

This system, called New Spelling (or Nue Speling), strives to maintain a regular correspondence between letters and phonemes – /ð/ (the *th* of *then*) is always *dh*, /eɪ/ (the *a* of *mate*) is always *ae*, and so on, though it is not completely rigorous about this since it allows variant spellings for unstressed vowels; in the above passage, *longger* ends with the same sound as *familiar*, but one is spelt *er* and the other *ar* (because of the link with *familiarity*). As well as trying to keep the correspondences regular, it also tries to keep them simple and therefore it largely eschews the various devices in traditional orthography whereby one letter tells the reader something about the pronunciation of another, such as the 'silent' *e* in *kite* and the double *p* in *hopping.* As a consequence, only a minority of words remain unchanged; a passage in New Spelling is readable but it looks decidedly strange.

A more pragmatic approach is taken by Dr Axel Wijk in his Regularized Inglish (Wijk 1969). He is prepared to allow one symbol to represent more than one phoneme, or one phoneme to be represented by more than one symbol, so long as this does not cause confusion. He identifies those patterns that occur most frequently in the vocabulary – the 'regular' ones – and alters the exceptions to bring them into line. For example, he allows *a* to stand for /æ/ (*mat*) or /eɪ/ (*mate),* but *want* and *any* have to be changed to *wont* and *eny.* The result is a system in which the

sound-spelling correspondences, though not as simple as in New Spelling, are regular and rule-governed, so that both native speakers of English and foreigners could have some confidence in using written English, the former in using their knowledge of pronunciation to guide their spelling, the latter in using their knowledge of the spelling to guide their pronunciation. In contrast to New Spelling, fewer then ten per cent of the words of English need to be changed, as can be seen in this example:

> We instinctivly shrink from eny chainge in whot iz familiar; and whot can be more familiar than the form ov wurds that we hav seen and written more times than we can possibly estimate? . . . At the first glaance a passage in eny reformd spelling looks 'queer' or 'ugly'. This objection iz aulwayz the first to be made; it iz perfectly natural; it iz the hardest to remoove. Indeed, its effect iz not weakend until the new spelling iz no longer new, until it haz been seen offen enuff to be familiar.

A similar pragmatism informs a more recent proposal, called Cut Spelling (Upward 1992). The idea is to remove redundant letters – *hymn* becomes *hym, kneel* becomes *neel, people* becomes *peple, apple* becomes *apl,* for example – on the grounds that these letters cause a lot of trouble and, since they perform no useful function, they will not be greatly missed. A few substitutions are also made, including *j* for *dg* (so *edge* is spelt *ej*) and *f* for *gh* (so *cough* is spelt *cof).* About ten per cent of letters disappear, and this is achieved, it is claimed, without excessive disruption to the appearance of more than a handful of words. Here is the example passage once more:

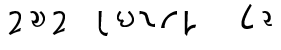
> We instinctivly shrink from any chanje in wat is familir; and wat can be mor familir than th form of words that we hav seen and ritn mor times than we can posbly estmate? . . . At th first glance a passaj in any reformd spelng looks 'queer' or 'ugly'. This objection is always th first to be made; it is perfectly natrl; it is th hardst to remove. Indeed, its efect is not weaknd until th new spelng is no longr new, until it has been seen ofn enuf to be familir.

In contrast to these three systems, which confine themselves to the Roman alphabet, two other systems of recent decades depart from it, the first modestly, the second flagrantly. Both aroused a lot of interest in their day. The first of these, the Initial Teaching Alphabet, was not proposed as a reformed orthography for general use but specifically as a medium for teaching reading and writing. Invented by Sir James Pitman, grandson of the Victorian spelling

reformer, it was introduced as an experiment in the early 1960s and by 1966 was being used in nine per cent of infant schools in England and Wales (Warburton and Southgate 1969). More than seventy publishers were producing schoolbooks in it (Pitman and St John 1969) and it looked set to gain widespread acceptance among teachers but, despite achieving positive results, it fell out of favour and is little used today. About twenty characters were added to the Roman alphabet so as to have enough letters to maintain a regular sound-symbol correspondence. Not all of these extra characters were brand new; several were formed just by linking two ordinary letters, like ʊu to represent /aʊ/ (the *ow* of *how).* The idea was that children would learn the basic principle of alphabetic reading and writing for the first year or so without being confused by the quirks of traditional orthography and then they would be led gradually into standard spelling. The extra characters were designed to be suggestive of common letter patterns. Here is how the phrase 'the initial teaching alphabet' looks in the Initial Teaching Alphabet:

ðe iniʃhal tæĉhiŋ alfabet

The other non-Roman system is the Shaw alphabet. George Bernard Shaw was a vigorous campaigner for spelling reform, arguing that the superfluous letters in traditional orthography were a terrible waste of time for writers – he did his own writing in Pitman's shorthand. He saw no future in rearranging the Roman letters, as other reformers have tended to, since he felt that the results would inevitably strike an ordinary reader as the work of an illiterate.[4] He left instructions in his will for the creation of a new alphabet for English, strictly on the principle of one letter for one phoneme. A competition was held, a winner eventually chosen from over four hundred entrants, and one of Shaw's plays, 'Androcles and the Lion', published in both the new and the old alphabets, side by side (Shaw 1962). He did not imagine that the new alphabet would replace the old one, but that the two would coexist, rather like Arabic and Roman numerals (MacCarthy 1969b). Here is the name 'George Bernard Shaw' written in the Shaw alphabet:



The case for reforming the orthography along alphabetic lines has counted many eminent linguists among its supporters, especially in the early part of this century. Classical philologists of the nineteenth century tended to regard modern languages such as English as degenerate forms of ancient languages, and it was partly as a reaction against this view that the modern science of linguistics emerged, asserting that modern languages were deserving of study in their own right. Allied to this was the feeling that the spoken language was the real language and that the written form was merely a somewhat inadequate representation of the spoken form (Sweet 1876). If the task of writing is simply to represent speech, then English spelling is a poor system, hence the linguists' support for spelling reform.

Since the Second World War, however, linguistic opinion has shifted. Writing is no longer seen merely as a reflection of speech, but as an alternative means of linguistic expression; not just a mirror of the speech system, but a system in its own right. Its primary purpose is not to show how words are pronounced but to convey meaning to the (generally silent) reader – 'to speak quickly and distinctly to the eyes'.[5]

That written language is not just speech written down becomes obvious when you realize that writing includes symbols that have no direct relationship to pronunciation. The *$10* of *It costs $10,* for instance, does not represent the sound of *ten dollars* – compare *Ça coute $10,* or *Das kostet $10.* The apostrophe conveys something to the eyes for which there is no counterpart in speech in *the girl's dresses* and *the girls' dresses.* In fact punctuation in general, though it performs some of the functions of intonation in speech, is a quite different sort of system for which there is no direct spoken counterpart.

Defenders of English spelling (Albrow 1972, Venezky 1970) point out that it is sometimes a positive virtue to depart from the pronunciation. Consider, for example, *cats* and *dogs.* The plural marker – the thing that turns *cat* to *cats* and *dog* to *dogs* – is different in pronunciation; it is /s/ on the end of /kæt/ but /z/ on the end of /dɒg/. A consistent alphabetic system would reflect this in the spelling, perhaps writing *kats* and *dogz.* But in terms of meaning, you are doing the same thing to *cat* as you are to *dog* – you are

adding a plural marker. We could, if we chose, indicate the plurality in writing by attaching some arbitrary symbol – say *2* – to the end of the word, thus writing *cat2* and *dog2.* The *2* here is not meant to correspond to any particular sound, any more than an apostrophe does; it just means 'plural form'. It is not indicating a phoneme; it indicates an element of meaning, which linguists call a *morpheme.* Well, up to a point, this is what English orthography does in using the letter *s* to mark the plural, regardless of whether the pronunciation is /s/ or /z/. It is a mistake to regard the *s* of *dogs* as a funny way of representing /z/; it is, rather, the standard marker of plurality.

The past tense of the verb provides another example of this. Suppose we used the symbol < to indicate past tense. *Jumped, crawled* and *landed* would be written *jump<, crawl<* and *land<.* Up to a point, this is what English orthography does in using *ed* as a past tense marker, regardless of the pronunciation. The *ed* is not a funny representation of /t/ (*jumped),* /d/ (*crawled*) and /ɪd/ (*landed);* it is the standard marker of past tense.

In these examples, English orthography is not being perverse in abandoning the alphabetic principle. There is something to be said for keeping a consistent representation for a given morpheme, and, if you choose to do this, you cannot also keep a totally consistent representation for phonemes. For complete consistency of the morpheme representation, we ought to write *mice* as *mouses* and *bred* as *breeded,* but when the forms are irregular (like *mice* and *bred),* the alphabetic principle reasserts itself and the orthography abandons the morpheme in favour of the phonemes.

Another feature of English spelling is to have sets of words looking the same if they are related in meaning, even though they may not sound the same, such as *photograph, photography* and *photographic,* where the pronunciation of the vowels varies because of the placement of the stress (Chomsky 1970). The silent *b* of *bomb* arguably helps to show its relationship to *bombard;* likewise the *g* of *sign* and *signature* and the *c* of *muscle* and *muscular* (Venezky 1970). The *a* of *polar* and the *o* of *author,* though they are pronounced the same, show the relationships of these words to *polarity* and *authority* (Haas 1969).

Chomsky and Halle (1968) suggest that a word – take *nation* as an example – is stored in the mental lexicon in a certain form and

that a speaker generates the appropriate stress pattern for a derived form, such as *national* or *nationality,* by applying a set of rules to this underlying form. These underlying forms are, obviously, not open to direct inspection, so it is a matter of conjecture what they look like (so to speak), but Chomsky and Halle argue that one can infer what the underlying forms must be, assuming that their theory is correct. It turns out, according to their theory, that their representation of the underlying form of a word often resembles the spelling more than the pronunciation. It is as though the orthography represents the underlying forms directly, and it is for this reason that they claim that the orthography is 'near optimal'.

There are other ways in which abandoning the alphabetic principle enables the orthography to make things clearer for the reader. It becomes possible to distinguish between homophones – words that sound the same but have different meanings, such as *bear* and *bare* or *there* and *their.* The final sound of *handed* and *fated* is the same as that of *splendid* and *fetid,* but the *ed* indicates that the first pair are inflected forms of verbs whereas the *id* marks the second as adjectives (Albrow 1972).

Word length is an important cue to the eye (Smith 1980a), and the orthography uses this by reserving the distinctiveness of two-letter words almost exclusively for function words – *in, be, by, do* and the like. (*The* would also be a two-letter word if we had not lost the letter Thorn.) Content words that would, on alphabetic grounds, be two-letter words usually get lengthened either by doubling – *inn, egg, ebb, bee* – or by the addition of an *e* – *bye, axe, doe, ore* (Albrow 1972).

It can be argued (Smith 1980b) that the peculiar spellings of words taken from other languages serve a function in indicating to the reader the foreign origins of the words and perhaps giving a hint as to the meanings. How do you pronounce *sopracella* and what is it? (I've invented the word.) If you felt, as you well might, that the word had an Italian look to it, then you might reasonably pronounce the *c* as in *cello,* even though this pronunciation of a single *c* is most unusual in English, and you might guess that it had something to do with music or Italian food.

One writer even rises to the defence of the silent *b* in *debt* (Sampson 1985), arguing that it makes the word more visually

distinctive. Whereas *det* looks rather like *bet, net* and *den, debt* doesn't. So silent letters and the like help the reader by making words more different in appearance than they are in sound. (Taken to extremes, this line of argument would presumably favour *caqt* for *cat* and *doxg* for *dog,* or, more distinctive still, *qqq* and *xxx?*)

The case for the defence so far has been that the alphabetic principle is not so self-evidently paramount that the orthography should follow it slavishly; there can be good reasons for departing from it. But the defenders have another point to make, namely that the orthography does actually stick more closely to pronunciation than its detractors would lead you to believe (Hanna et al. 1966, Venezky 1970, Wijk 1966). It does not do so, admittedly, in the simplest one-letter-one-phoneme way, but it has other ways of indicating pronunciation.

Consider the final *e* of *fate, village* and *mice.* Whatever functions these e's had in former times, they now indicate the pronunciation of the preceding vowel *(fate)* or consonant *(village)* or both *(mice).* A strict letter-phoneme analysis dismisses *fate* as an irregular spelling – the *a* is not pronounced /æ/ as in *bat* and the *e* is silent – but it can be argued that, providing you understand the function of the final *e, fate* corresponds perfectly well to the pronunciation. In fact, if you are prepared to consider the effect that letters have on each other, many of the apparently irregular spellings are seen to be reasonably rule-governed. An initial *c* before a vowel, for instance, can be pronounced as /k/ or /s/, but this is not a serious source of confusion; if it's before an *e, i* or *y,* it's /s/ (*cement, cinder, cyanide*), otherwise /k/ (*cat, cot, cut).*

There are, admittedly, genuine exceptions. The word *of,* for example, is the only word in which the letter *f* corresponds to the phoneme /v/ (Carney 1994). The spellings *one* and *two* are highly peculiar; *could, would* and *should* are pretty strange. It is unfortunate that so many of these oddly spelt words are also common words. Taking the whole vocabulary, it is possible to maintain that the great majority of spellings are orderly, providing you accept that the rules are more complicated than one-letter-one-phoneme (Wijk 1966).

A feature of English orthography which many people respond to without being consciously aware of it is the doubling of consonant letters to mark stress in words of a certain pattern – compare *cinema*

with *dilemma* (Smith 1980b). Here again, a seemingly redundant letter (why use *mm* when *m* will do?) is actually performing a function (Smith 1980a).

Position is another factor which the simple letter-phoneme analysis ignores; a letter may stand for a certain phoneme in one position in a word but not in others. George Bernard Shaw joked that *fish* could be spelt *ghoti* in English – *gh* as in *cough, o* as in *women* and *ti* as in *nation.* But it couldn't (Haas 1970). A *gh* at the beginning of a word is always pronounced /g/; it's only pronounced /f/ in the middle or at the end. A *ti* corresponds to /ʃ/ only at the beginning of certain syllables such as *-tion, -tial, -tious* (*potion, martial, bumptious);* a *ti* at the end of a word would be /tɪ/ or /ti:/. And the correspondence of *o* to /ɪ/ as in *women* is extremely rare.

A further point in defence of the traditional orthography is that, quirky though it may be, it is at least standard for all users of the language. There are a few areas of uncertainty – examples include *gaol* or *jail, benefited* or *benefitted, slow-worm* or *slowworm, connection* or *connexion, organise* or *organize* – but there is only one accepted spelling for the great majority of English words, and even British-American differences are relatively few. (The last of these examples – *-ise/-ize* – is not a clear British-American split, as many people think; the *Oxford English Dictionary* favours the *z* form.)

It is hard to see how standard spellings could be maintained in a purely alphabetic system, since people's spelling would, to some extent, reflect their accent. Suppose that a reformed system used *a* for /æ/ and *aa* for /ɑ:/, so that *gas* and *palm* were written *gas* and *paam.* Speakers of Southern British English would spell *pass* as *paas* (more like *palm* than *gas*) whereas people from the North of England would spell it *pas* (more like *gas* than *palm*). A similar problem occurs with *cloth, lot* and *thought.* British speakers make the same vowel sound for *cloth* and *lot,* contrasting it with *thought.* Americans, however, make the same vowel for *cloth* and *thought,* and a different one for *lot.* (In fact their *lot* sounds like their *palm.*) Scots, Irish and most Americans would feel the need of an *r* in the word *source* (for them, it doesn't rhyme with *sauce*), but most English people wouldn't. A Jamaican spelling in an alphabetic way would write *pot* and *rot* as *pat* and *rat.* English is an international language and there are many more examples of such differences

from around the world (Wells 1986).

This criticism does not apply to all reformed orthographies. It will be obvious from the examples I gave earlier that, while sharing the aim of bringing English spelling closer to pronunciation, they do not all insist on the principle of one letter per phoneme. When writing in one of those orthographies that do insist on the principle, however, a writer has some decisions to make. Some words have 'strong' and 'weak' pronunciations; for instance, should *and* be written as if it rhymed with *Ann* or like the middle syllable of *fish and chips?* Which pronunciation (and therefore spelling) should be chosen for such words as *controversy, azure, subsidence, patent* and *laboratory?*[6] The sentence I gave earlier from the Gettysburg Address was written with an English rather than an American accent (Americans might prefer *nuu* to *nue* for *new,* for instance) (MacCarthy 1969a). Bernard Shaw anticipated this difficulty by instructing that the transliteration into his new alphabet should assume a pronunciation 'to resemble that recorded of His Majesty our late King George V' (to which he added, mysteriously, 'and sometimes described as Northern English' (Shaw 1962)). Would an alphabetic orthography oblige us to adopt a standard pronunciation for writing or at any rate for publication?

Defenders of the orthography have been in the ascendant in recent years; the Kingman Report on the Teaching of English Language, for example, lays some emphasis on the patterns and regularities in English spelling and states that, by the age of seven, children should 'understand that spelling obeys rules' (Kingman 1988). But the spelling reformers have not allowed the defenders' claims to go unchallenged.

The defenders point out how traditional orthography keeps function words short and makes other words longer; the reformers reply that it does not do this consistently: *ox* has only two letters but it is not a function word (likewise *ax* in America); *could, would, should* and *ought* are all function words but their spelling is needlessly long. The defenders claim that the ending *ed* marks verbs whereas the ending *id* marks adjectives, but where does that leave *naked* and *wicked*? The defenders show how the double *l* marks the pronunciation of the vowel in *doll* as against *dole,* but it doesn't seem to work with *role* and *roll* or *pole* and *poll.*

The defenders can argue that the doubling of consonant letters serves a function in indicating the short vowel and strong stress of the preceding syllable, in words like *umbrella* and *antenna.* The reformers reply that words of this type constitute no more than a small island of regularity in a sea of confusion. An adequate general description of when the doubling of a consonant letter does and does not occur is inordinately complicated. A recent attempt (Carney 1994) runs to eleven 'rules', some of them requiring the application of linguistic concepts such as 'Latinate prefixes' which can hardly be assumed to be part of the intellectual equipment of the ordinary user of English (to account for the doubling in words like *commit* and *illicit*). And even then there are exceptions – *comic/traffic, solid/pallid, radish/rubbish, habit/rabbit, robin/bobbin, planet/sonnet, treble/pebble* – and many more.

The defenders like to display sets of words that keep the same spelling despite differences in pronunciation, such as *nation* and *national;* the reformers reply that there are many sets of words that don't: *fire-fiery, high-height, speak-speech, space-spatial, aeroplane-aircraft, comparison-comparative, proceed-procedure* and many more. In fact an analysis of about six thousand of the commonest words in English suggests that pairs of words like *nation* and *national* are not particularly characteristic of English orthography but are just examples of a pattern that can be seen in a number of words, and not a notably large number at that (Yule 1978).

The defenders emphasize the help which is given to readers by distinguishing in spelling between words that are pronounced the same (homophones, such as *fowl* and *foul*), but it is debatable whether readers need any help with this. There are many words in English which are both pronounced and spelt the same and this does not seem to cause confusion – *bank* (river/money), *tender* (soft/contracts/steam engines) and many more. If the words are sufficiently distinct in meaning and perhaps also in part-of-speech, they are unlikely to be confused; are *heal* and *heel* in such danger of confusion that they need to be spelt differently, as compared with, say, *ball* (cricket) and *ball* (Cinderella)? Conversely, traditional orthography sometimes gives the same spelling to words pronounced differently, thus creating homographs such as 'her *tears* of joy' and 'she *tears* her hair out,' and this does occasionally cause confusion. Take the sentence, 'We intend to ask people

about the magazines they read and the comics they enjoyed when they were young.' Does *read* rhyme with *reed* or *red;* are we talking about the magazines they read now or the ones they used to read? A careful writer would have to rephrase the sentence solely to get round this flaw in the orthography.

And so the argument goes on. To sum up, the critics' charge is simple: English spelling is so far out of line with pronunciation and so riddled with inconsistencies that it creates needless difficulties. In the words of the French lady whom I quoted in the introduction, 'There are no rules in English, only exceptions.' Schoolchildren and their teachers, foreigners learning the language, writers and printers, in fact just about everybody who uses written English would benefit from an orthography that was closer to being consistently alphabetic. The defenders reply that English spelling is, in fact, closely related to pronunciation, but not in a straightforward way. They almost turn the French lady's remark on its head, arguing that there are few real exceptions in English spelling, only lots of rather complicated rules. Besides, they say, it is a disservice to English spelling to regard it merely as a poor alphabetic system. Though basically alphabetic, it cannot mirror pronunciation exactly since it is trying to do other things as well, such as maintaining a consistent spelling for morphemes.

Both sides, I think, overstate their case. As one writer has observed, the judgement whether English spelling is a basically regular system containing irregularities or a basically irregular system containing certain patterns is not a statement of fact but of attitude; the same bottle can be described as half-full by one man and half-empty by another (Upward 1988). It is naive to parade endless lists of inconsistencies between spelling and pronunciation, as reformers have been inclined to do. The inconsistencies have various origins and various functions, and some of them are arguably a help to the reader. On the other hand, the systems of rules, sub-rules and exceptions that defenders of the orthography have devised seem unreasonably complicated. There is something ostrich-like about maintaining that the system is orderly when a great many people clearly do not find it so.

Two observations may be made from the point of view of spelling correction. The first is that people's preferences vary; the very existence of this long-running dispute between reformers and

conservatives suggests that a single orthography, reformed or not, would almost certainly not suit everyone. An American psychologist divides people into 'Phoenicians' and 'Chinese' (Baron et al. 1980). The 'Phoenicians' (so called because the Phoenicians are credited with inventing the alphabet) are adept at using the relationships between spelling and pronunciation. They are good with nonsense words ('Twas brillig and the slithy toves'), either reading them aloud or producing plausible spellings for nonsense words read out to them. They cope better with regular spellings such as *lunch* and *ship* than with irregular ones like *island* and *yacht.* With the 'Chinese', it's the opposite; they seem not to make much use of sound-spelling relationships, so they are not much good with nonsense words but they are not upset by irregular spellings – they do not have a lot more trouble with *yacht* than they do with *ship.* Further evidence of such differences comes from an experiment in which students were given a list of words and were asked to say how 'rational' the spelling of each one was, in their opinion, and to suggest alternative spellings for those they considered to be not completely rational. Some students altered almost all the words while others changed only a few (Baker 1980).

The second observation is that, whatever advantages we gain by departing from the alphabetic principle, it is when we are reading, rather than writing, that we feel the benefit. When writing, most of us would surely be happier with a more consistently alphabetic system.

A possible compromise would be to retain the standard orthography for reading, but for each person to use his own preferred orthography for writing. While people write straight onto paper, this is not possible since the marks that the writer makes are the very marks that the reader reads. But it is, in a way, just what a computerized spelling corrector offers: you use whatever spelling system seems to you most natural for writing, and the computer converts it into the standard system for reading. Looked at like this, it is not so much correcting wrong spellings into right ones as simply converting one system into another.

Notes

1. The words 'vowel' and 'consonant' can be a source of confusion. They properly refer to phonemes. Every phoneme is unambiguously a vowel or a consonant; the word /bɔ:t/ (*bought*), for example, has one vowel and two consonants. But the words are often used to refer to letters – *bought* (the spelling) has two vowels and four consonants. Letters are not unambiguously one or the other. The letter *y* is sometimes a consonant (*yellow*) and sometimes a vowel (*nylon*); the vowel letter *u* corresponds to the consonant phoneme /w/ after *q (queen, quick);* the consonant letter *w* forms part of the representation of a vowel in *bowl.* I will try to make it clear whether I am talking about letters or phonemes where the distinction is important.
2. This is a considerable problem for linguists trying to map the sound-to-spelling correspondences in English; see pages 32-48 of Carney (1994).
3. Quoted by Scragg (1974).
4. He puts this view in a letter to *The Times* dated 25 September 1906, quoted by Sir James Pitman (1969).
5. This is a quotation from a Czech phonetician, Frinta, 1909, cited in Vachek (1973).
6. See *Notes on the spelling* by Peter MacCarthy in 'Androcles and The Lion' (Shaw 1962).

CHAPTER FOUR

A corpus of spelling errors

When people are parodying the writing of poor spellers, they tend to write such things as 'Down wiv skool', but in fact joke-misspellings such as *skool* form only a minority of misspellings, even in badly spelt work. What sort of misspellings do people actually make?

Opinions may vary about what constitutes a representative sample of spelling errors, but I imagine most people would mean errors made by a cross-section of adults rather than by children or by a particular group such as university students, and in free writing rather than in spelling tests or psychological experiments. Unfortunately there is, to my knowledge, no such corpus available. Out of a number of collections of errors that I have gathered for my research (Mitton 1985), the one I am about to describe is the nearest to 'errors in general'. It is a large collection taken from short compositions handwritten by fifteen-year-olds.[1] I am not sure how far this can be taken to be representative of the work of adults. It has been shown that people's reading continues to improve after they have left school (Rodgers 1986), but it is not known whether their spelling also improves and, if so, by how much.

The original material was gathered in 1970 by Dr Margaret Peters as part of her research into the teaching of spelling (Peters 1970). She visited all the secondary schools in Cambridge (U.K.) and gave a spelling test to pupils of school-leaving age (then fifteen years of age). After the test, she asked them to write for ten minutes on the topic 'Memories of my primary school'. Pressure of work prevented Dr Peters from analysing the data, but she kindly gave the scripts to me.

I had all the spelling errors from 925 of these compositions keyed into the computer. The collection contains slips (the writer knew the correct spelling but wrote something else) as well as wrong spellings (the writer did not know the correct spelling). Although it is often fairly clear whether an error was a slip or a wrong spelling, it is not possible to distinguish between them in all cases.

One composition was completely unreadable. The remaining 924 had a mean length of 184 words. The number of recorded errors was 4218, a rate of twenty-five per thousand words. Eleven per cent of the compositions had no errors in them; at the other extreme were a few so badly spelt as to be almost incomprehensible. The best spellers therefore contributed no errors at all, the worst spellers contributed large numbers, so the corpus consists largely of errors produced by the poorer spellers. The following three examples illustrate, in order, the average-to-good, average-to-poor and very poor:

(1) It was a small school situated on the outscirts of Cambridge. There was a large field behind where we held our sports and at dinner hours played football or in summer cricket. There were not many pupils, so it was rather quite at times. There were mostly women teachers although I clearly remember Mr. Heron who occassionly took us canoeing on the river. Each day we had a french lesson and we used cartoons, and we also acted these out. I had great pleasure in this, as it was a change from every day school work. We had many plays, a may play, a christmas play and other ones during the year. The may play particularly stands out. We used a cricket net and this was full of flowers intertwined about the netting. There was a long grand procession, and fortunately it was a very beautiful day. I can always remember our sports day, as it was one of the special occassions in the year. There was rosettes, and drinks although they were not free. We had the traditional sports, the sack race etc. At school we used to do modelling, making planes which most of us did. We used to try them out on the field but many were unsuccesful. In the early part of my primary school we used to go to a seperate building, not far away and situated behind a church.

(2) At my primary school our head master was Mr skart he was very nice. Me and my friend Doreen yourst to wash the teaches cup up after school. Every year we had a sports day I was in Yellow and Blue teams. The first Second and Firld year I was in Blue. The forft year I was in Yellow. I enteard a panting compertison I didnot come any were. But every body who enteard got a packet of sweets. I sat near my friend Doreen in class. When I was in the firld year our class would lean somethink to say in the hall and we would

say it in the hall. For a tearm I had french. I did high Jump some dinner times. Every week we had to copey a pitcher then pant it. Neally every year I was in the hope race at storts day and also others.

(3) That my school I live it and a men come to the school and tine to tiece me to rend and he life and than a lade and the lade life as will so I that to tine and rend my still. Mr home was are techer and he was needs and we play game. We did feirne. And we did and History and nut ball and roundosy, and I reant in juan it. and Be for I life are pase me 14 summer pasecart and I was ferod paret indored. and Mr home cave same Boys and girl 6d for pasecart.

A possible interpretation of the third passage is as follows:

... ??? ... and a man came to the school and tried to teach me to read and he left and then a lady and the lady left as well so I had to try and read myself. Mr Home was our teacher and he was nice and we played games. We did French. And we did History and netball and rounders. And I really enjoyed it. And before I left I passed my 14 [yards] swimming pass-certificate and I was very proud indeed. And Mr Home gave some boys and girls 6d [sixpence] for the pass-certificate.

The last passage illustrates not only the poor standard of the poorest compositions but also the extent to which the poor spelling was specifically to blame for the unintelligibility. Poor hand-writing, punctuation and syntax usually accompanied poor spelling, but it was the spelling that made them unreadable. With only the spelling corrected, most of these passages would become reasonably comprehensible.

Real-word errors

To anticipate later chapters for a moment, a type of error that is especially troublesome for spellcheckers occurs when a writer writes some other word in place of the one intended, such as *forth* for *fourth.* I call these 'real-word errors'. They are surprisingly common.

Forty per cent of the errors in the corpus were real-word errors. Table 4.1 presents a subdivision of the real-word errors and also separates errors on function words from errors on content words.

Table 4.1 Real-word and non-word errors

	Function w'ds		*Content w'ds*		*Total*	
Real-word err's:						
– wrong word	382	45%	351	10%	733	17%
– wrong form	42	5%	350	10%	392	9%
– word-divis'n	127	15%	398	12%	525	13%
Half real word	9	1%	40	1%	49	1%
Non-word	295	34%	2224	67%	2519	60%
Total	855	100%	3363	100%	4218	100%

Function words are words like *the*, *to* and *of*, which indicate the grammatical structure of a sentence but which carry little meaning in themselves; the list of about three hundred function words that I used for this analysis is given in full in Appendix Two.

These figures for the overall incidence of real-word errors can be compared to those obtained from three other studies of handwritten material – a study of examination scripts submitted for the Cambridge University entrance examination (Wing and Baddeley 1980) and two other studies of schoolchildren's compositions (Sterling 1983, Brooks et al. 1993). Since some of these studies ignore word-division errors, we first have to exclude word-division errors from the above table, which gives a figure of thirty-two per cent for the incidence of real-word errors. The figures from the three studies are twenty-six per cent, twenty-six per cent and twenty-nine per cent respectively.[2] It appears that real-word errors account for about a quarter to a third of all spelling errors, perhaps more if you include word-division errors.

Wrong-word errors

The first group of real-word errors – wrong-word errors – are those where some other word was written in place of the right one.

Occasionally it was, presumably, only by chance that the misspelling matched a dictionary word, as in *pict* for *picked* or *tort* for *taught.* Generally the misspelling was some other word from the student's vocabulary, such as *sought* for *sort* or *know* for *now.* Table 4.2 lists some of the most common wrong-word errors.

The students were more prone to make these errors when writing function words; in fact, over half of all the wrong-word errors were on function words. These were often errors on short, very familiar words, as in 'you we treated like babies,' and 'he name was Mrs Williams.' In eighty per cent of the wrong-word errors on function words, some other function word was written in place of the one intended. I say more about these errors in Chapter Six.

Table 4.2 Some common wrong-word errors in the corpus

Word A	*in place of Word B*		*vice-versa*	*Total*
to	too	28	10	38
were	where	19	19	38
of	off	22	4	26
their	there	17	4	21
forth	fourth	19	0	19
quiet	quite	14	5	19
are	our	18	0	18
know	now	8	1	9
aloud	allowed	8	0	8

Each figure shows the number of times this particular error occurred in the corpus.

Wrong-form-of-word errors, inflections, apostrophes

The second group of real-word errors consisted of a wrong form of the word rather than a completely different word, such as present-tense verbs in place of past ('the best thing I like was to play,' 'eventually the time arrive when ..'), or singular nouns in place of plural ('dozens of thing I could say,' 'five other primary school'). *Use to* for *used to* was easily the most common single error of this

kind, accounting for sixteen per cent of these wrong-form-of-word errors, though the high frequency of this phrase, and therefore of this error, is obviously related to the topic of the compositions (memories of primary school). Many of these errors were presumably slips, though not necessarily all of them – it is possible that some of the writers of *use to* thought that this was correct.[3]

Inflected forms gave rise to many errors, real-word and otherwise. Table 4.3 shows the types of error made on nouns with a plural *s* or *es*, verbs with a past-tense or a past-participle *ed*, *d* or *t*, and verbs with a present-participle *ing*. The table includes only those words where the error was confined to the inflection (so *lavatorys* would be included but not *lavertrys*). Three-fifths of these errors were real-word (i.e. wrong-form-of-word) errors, the others non-words.

Table 4.3 Some inflection errors

	Noun+s	*Verb+ed*	*Verb+ing*
Inflection omitted	54%	71%	10%
Base form not adjusted	23%	11%	64%
Base form wrongly adjusted	23%	18%	26%
Total (= 100%)	187	178	61

Some of the inflection errors involved apostrophes, such as 'memory's of primary school'. Errors involving only the omission or misplacing of an apostrophe (*the boys cloakroom, the boy's cloakroom*) were generally not included in the corpus. Inspection of a sample of 110 of the compositions revealed 57 errors over apostrophes. The same compositions contained only 70 correct apostrophes, so these students were getting the apostrophe wrong almost as often as they were getting it right. The most common mistake was to omit it, though it was also sometimes misplaced or incorrectly inserted.

Word division

The third subset of real-word errors were words incorrectly divided into two in such a way that both parts were real words, as

in *my self* and *in side.* A particularly inventive one was *head miss dress* ('The head miss dress was called Mrs Charles.') These accounted for thirteen per cent of all the misspellings. In two other studies of misspellings which resemble mine in treating word-division errors as a separate category, the incidence of real-word errors was remarkably similar – twelve per cent in one case and thirteen per cent in the other.[4]

English usage is somewhat variable over word division. *Boot laces, boot-laces* and *bootlaces,* for example, all seem acceptable. Some of the word divisions that are counted as errors in the corpus – perhaps as many as half of them – are marginal. Even those that would be widely agreed to be errors are generally not serious – presumably no-one would read 'in side the room' as anything but 'inside the room' – but the versions with and without a space do not always mean the same. Consider 'every body in the mortuary burst out laughing.' (This example does not come from the corpus; I've invented it to make the point.) Pedants who insist that *alright* should be spelt *all right* are mistaken, in my view – 'Your answers were all right,' simply does not mean the same as 'Your answers were alright.'

In about a tenth of the word-division errors, part of the word had been respelt to form a word or to form a different word – *there for, all ways, sum times, know one.* The small group of errors called 'half real words' were word-division errors where one of the halves (the first half in seventy per cent of them) was a real word and the other half a non-word, such as *to gether* and *evry body.* (There were just five divided-word errors where both halves were non-words.) Some of the more common word-division errors (related, obviously, to the topic of the compositions) are listed in Table 4.4.

The opposite error – of joining two words when they ought to be separate, as in *infront* – was less common; less than three per cent of all the misspellings were of this type, the most common single one being *alot* which occurred thirty-seven times. (It should be remembered that this material was handwritten; it could well be more common to run words together in typewritten text.) This type of error rarely produced a real word.

Table 4.4 Some common word-division errors

class room(s)	51
play ground(s)	23
out side	22
some times	10
every body	8
all ways	6
every one	6
my self	6

Each figure shows the number of times this particular error occurred in the corpus.

Simple errors

The designers of spelling correctors have leaned heavily on two findings from earlier analyses of errors. The first is that misspellings are generally correct in the first letter (Pollock and Zamora 1984, Yannakoudakis and Fawthrop 1983a). My analysis confirms this; only seven per cent of the misspellings were wrong in the first letter. These errors were often caused by 'silent' initial consonants, as in *know* and *write.*

The second finding from earlier analyses (Damerau 1964, Pollock and Zamora 1984) is that over eighty per cent of misspellings are single-error misspellings, i.e. misspellings that differ from the correct spelling in just one of the following ways (the correct spelling here being *albatross*):

one letter omitted	albtross
one letter wrong	akbatross
one letter inserted	alabatross
an adjacent pair transposed	ablatross

Only sixty-nine per cent of the misspellings in the corpus were of this type. The reason for the higher proportion of multi-error misspellings in this corpus is that it was collected from people of all levels of spelling ability, with a full representation of poor and very

poor spellers. As I will show later, poor spellers tend to make worse misspellings.

Homophones

If spelling errors were generally the result of people producing spellings by some sound-to-letter system, most misspellings should be homophones of the correct spellings, such as *skool* for *school.* To what extent does the corpus bear this out?

Deciding whether a misspelling does or does not sound like the right word is not completely straightforward, especially for non-word errors. It depends on how you divide the word, whether or not you apply certain rules, and whether you derive a pronunciation by rule or by analogy with other words. If you divide *biger* as *big+er* and apply a simple letter-sound mapping, then *biger* could be a homophone of *bigger* – and it may be that the students who wrote it thought it was – but applying slightly more complex rules changes the pronunciation of the *i* (*tiger*) or the *g* (*wager*) or both. The same applies to *beter, swiming* and many more. Similarly, *onece* is a homophone of *once,* and *youst* of *used* (in 'I used to'), if you divide them appropriately (*one+ce, you+st*) but not if you don't. An example of the analogy problem is *trambs,* a misspelling of *trams.* If you apply the simplest mapping then it's not quite a homophone, but it is if you read it by analogy with *lambs* and most other words ending *mb.*

A further complication is that people's pronunciation varies. *Where* and *were* are not homophones for me, but they are for some people, or very nearly so. And words that are not homophones in isolation can be in context, such as *have* and *of* in 'I might of done.'

I divided the errors, therefore, into three groups rather than two – homophones, near-homophones and non-homophones. Table 4.5 shows the division of the errors into these groups. The table includes all the non-word errors but only the first group of real-word errors (the wrong-word errors), i.e. it does not contain the wrong-form-of-word errors or the word-division errors.

Table 4.5 Homophone errors

	Wrong-word errors		*Non-word errors*		*Total*
	Funct'n	*Content*	*Funct'n*	*Content*	
Homoph's	8%	32%	44%	53%	44%
Near-hom's	34%	11%	13%	12%	15%
Non-hom's	58%	57%	43%	35%	41%
Total (100%)	382	351	295	2224	3252

The right-hand column of Table 4.5 shows that the full set of errors was fairly evenly divided between those that sounded like the correct word and those that didn't. Misspellings of content words were often homophonic non-words. In many of these the error was minor (*becides, misstake, extreamly),* often confined to the choice of representation for an unstressed vowel (*chocalates, untideness, proposterous, conveniances);* in others it looks as though the students were writing words whose spelling they were unsure of and they were falling back on some sound-to-spelling system (*vaig, aporators, teribal*). The same processes sometimes produced wrong-word-errors by accident, some of which were homophones (*lessen* for *lesson, story* for *storey*) but most of which were not (*manly* for *mainly, scarred* for *scared, exiting* for *exciting*). More often, however, wrong-word-errors seem to have been produced by the student writing a known spelling in place of the one required, often homophonic (*where* for *wear, court* for *caught, righting* for *writing*) but not always (*joiner* for *junior),* especially for function words (*they* for *then, and* for *an*).

A few misspellings were homophones or near-homophones of the right words even though they differed quite a lot in appearance (*fersilaty's* for *facilities, wisel* for *whistle*). But misspellings that sounded right or nearly right generally also looked like the right words (*pantemine, chrisanthimums*) and, conversely, many of those that sounded wrong also looked wrong (*rienind* for *reading, brtatons* for *potatoes*).

Passable and poorer spellers

As I mentioned in my description of the corpus, Dr Peters gave the students a spelling test before asking them to write the compositions. She read out each word, then gave the word in a short sentence, then read out the word again, after which the students wrote it down. Some of the results of this test are recorded in the file, in particular any misspellings of the following ten test words: *eye, fight, friend, done, any, great, sure, women, answer, beautiful.*

On the basis of these ten words, the students can be divided in a rough-and-ready way into 'passable spellers', who got all ten right, and 'poorer spellers', who got at least one wrong. By this definition there were 631 passable spellers and 293 poorer ones. The poorer spellers wrote shorter compositions – a mean of 127 words compared to the passable spellers' 210 – and their error rate was sixty-three per thousand words as against the passable spellers' fourteen. Though forming only about a third of the sample, the poorer spellers contributed over half the errors.

Tables 4.6 and 4.7 compare the errors made by these two groups. The pattern of their errors on function words was very similar, so these tables are confined to content words. As in the above analysis of homophones, Table 4.7 excludes the wrong-form-of-word and word-division errors.

Looking at the various types of error in terms of how serious they were, there was a sort of continuum between word-division errors (*in side, every one*), many of which were only trivially different from the correct spelling, and non-homophonic non-words (*feirne, ferod*), which sometimes bore no close resemblance to any word at all. Table 4.6 shows that more of the passable spellers' errors were of the more trivial kind, while Table 4.7 shows that more of the poorer spellers' errors were of the more serious kind. In short, the poorer spellers did not just make more errors; they also made worse errors.

Table 4.6 Passable and poorer spellers (content words only)

	Passable	*Poorer*
Real-word errors:		
– word division	17%	7%
– wrong form of w'd	10%	11%
– wrong word	8%	12%
Half real word	1%	1%
Non-word	64%	68%
Total (=100%)	1549	1814

Table 4.7 Homophone errors: passable and poorer spellers (content words only)

	Wrong-word errors		*Non-word errors*	
	Passable	*Poorer*	*Passable*	*Poorer*
Homophones	48%	23%	65%	42%
Near-homophones	12%	11%	12%	13%
Non-homophones	40%	66%	23%	45%
Total (=100%)	125	226	985	1239

Notes

1. The corpus described here is just one of a number of files of spelling errors which are available in machine-readable form for bona-fide research from the Oxford Text Archive (address on page 130).
2. Wing and Baddeley (1980) collected all the spelling errors in examination scripts of forty candidates applying for entrance to colleges of Cambridge University in 1976. Excluding those errors which the candidates themselves corrected, we obtain a figure of twenty-six per cent for real-word errors (134 real-word errors out of 513). I present more figures from this corpus in Chapter Six.

Sterling (1983) analysed 547 misspelt words taken from essays written by 56 twelve-year-old children attending a secondary school in Alloa, Scotland in the late 1970s. Excluding word-division errors ('splits', as Sterling calls them) gives 125 real-word ('lexical') errors out of 475 (twenty-six per cent).

Brooks, Gorman and Kendall (1993) analysed 3342 spelling errors taken from the first ten lines of 1492 short compositions written by representative samples of schoolchildren in England and Wales at the ages of eleven and fifteen collected between 1979 and 1988 by the Language Monitoring Project of the Assessment of Performance Unit. They found that twenty-nine per cent of the errors (962) were in real words. This figure is not directly comparable with the thirty-two per cent obtained from Table 4.1 since they counted errors rather than misspelt words – *empossable* for *impossible,* for example, would count as two errors (one for the *e* and one for the *a*). The distribution of errors per word might have been different for real-word errors than for non-words, but I doubt if this will have altered the results substantially.

3. This short and simple-looking word – *used* as in *used to* – gave rise to a surprising number of errors. The following table presents all the misspellings of this word:

Table 4.8 Misspellings of 'used'

use	66	you	4	uset	1
youst	18	us	3	ust	1
uses	7	yous	3	usued	1
us't	6	youwhile	2	yourst	1

4. In the Alloa study mentioned in note 2 (Sterling 1983), 72 of the 547 misspellings (thirteen per cent) were word-division errors.

Upward analysed 357 spelling errors (defined in such a way that one misspelling could contain more than one error, as in the Brooks study mentioned in note 2) taken from 73 sentence-completion questionnaires, with sentences such as 'The people I am happiest with are ... ,' filled in by 73 fifteen-year-old students in 1991 at a small-town comprehensive school in the East Midlands of England (Upward 1994). He found 44 word-division errors (twelve per cent). He also noted the strong tendency to turn *a lot* into one word – *alot.*

CHAPTER FIVE

Misspellings

Some spelling errors occur because the writer does not know how to spell the word; these are sometimes called 'errors of competence'. Others occur when the writer knows perfectly well how to spell the word but for some reason writes or types something else; these are 'errors of performance'. This chapter is about the first kind; the next chapter deals with the second kind.

Everyone knows that some words are harder to spell than others. Everyone knows also that hard words are not uniformly hard all along their length; they contain hard spots. What makes a hard spot hard?

The mistakes people make in spelling tests provide some answers to this question. Although spelling tests are artificial in the sense that people are asked to write words that they might not normally use in their own writing – in some cases words that are possibly not in their vocabulary at all – the results have the obvious virtue of presenting us with large numbers of attempts at the same words. Tables 5.1 and 5.2 give the misspellings of six words made in spelling tests by two samples of school-leavers (fifteen years of age).[1] Table 5.1 shows, for example, that seventy-two per cent of the students spelled *gallery* correctly while two per cent made no attempt at it; fourteen students produced the misspelling *gallary*, eight produced *galery* and so on.

These tables are typical of the results obtained from spelling tests. There is generally one popular misspelling, a handful of misspellings each produced by a few students, and then a long list of more-or-less bizarre efforts each produced by just one student. It is noteworthy that the great majority of the efforts, even some of

Table 5.1 Some misspellings of 'gallery', 'ventilated' and 'scissors'

gallary	14	ventelated	24	sissors	22
galery	8	ventalated	14	siccors	7
galary	5	venterlated	8	scisors	5
gallory	5	ventillated	2	siscors	4
gallerey	2	ventolated	2	sisers	3
gallry	2	ventulated	2	sissers	3
gaeroe	1	ventylated	2	scisers	2
galerry	1	dentilated	1	scissor	2
gallerie	1	vedulated	1	sisors	2
gallorry	1	venlatated	1	cezzous	1
gallowry	1	ventallated	1	cissuce	1
galory	1	ventearіated	1	saciarres	1
garley	1	vented	1	scicsors	1
garllry	1	ventelaind	1	scirrors	1
garrey	1	ventenlited	1	scisous	1
gary	1	venterelated	1	scisscors	1
		ventialed	1	scissers	1
		ventilate	1	scissocers	1
		ventilented	1	scors	1
		ventilt	1	secors	1
		ventlated	1	sessiors	1
		ventorlated	1	sicars	1
		venturlated	1	sicciors	1
		vetlettd	1	sicer	1
		vimleated	1	sicerse	1
		vintilated	1	sices	1
		wellvented	1	sicorrs	1
				sisions	1
				sisore	1
				sisorse	1
				sisscors	1
				sissor	1
				sissow	1
				sizers	1
				sizzors	1
				sliemer	1
				sorriors	1
Correct	72%		56%		51%
No attempt	2%		2%		5%
n (=100%)	172		172		172

Table 5.2 Some misspellings of 'exhibition', 'successful' and 'definite'

exibition	19	succesful	14	definate	42
exebition	4	succesfull	5	deffinate	7
exabison	2	sucessful	5	defant	2
effebishon	1	succsesful	4	defent	2
esidition	1	successfull	3	defente	2
exabion	1	sucessfull	2	definant	2
exabishon	1	sucsesful	2	diffinate	2
exabision	1	sussfull	2	deafernate	1
exabition	1	scucfull	1	deafnet	1
exbishion	1	scuksefuly	1	defanit	1
excbition	1	seccesful	1	defantnut	1
excibation	1	secessful	1	defenat	1
excibition	1	secsecfuly	1	defenert	1
exdishion	1	secsesful	1	defenet	1
exebechon	1	sesesful	1	defenite	1
exebesion	1	sexself	1	defernat	1
exebistion	1	succeful	1	defernate	1
exespan	1	succeshful	1	defferent	1
exhibation	1	successfuly	1	deffinite	1
exhibtion	1	succful	1	defienant	1
exibishtion	1	suceseful	1	definat	1
exidition	1	sucesfful	1	definent	1
exipition	1	sucksesful	1	definet	1
expane	1	sucksesfull	1	definete	1
expetion	1	sucsessful	1	defnent	1
expidian	1	sucsfully	1	defunet	1
		surseful	1	defunnet	1
		suskfull	1	desfient	1
				detinate	1
				dieffinate	1
				diffenent	1
				diffiant	1
				diffinant	1
				difinent	1
				dontf	1
Correct	55%		45%		20%
No attempt	2%		3%		1%
n (=100%)	110		110		110

the odder ones, are recognizably attempts at the words they are meant to be rather than at any other word. This observation, though obvious, is of some importance for spellcheckers since it suggests that, for most misspellings, it ought to be possible to make a good guess at the required word.

Silent letters are hard spots, as exemplified by the most popular misspellings of *scissors* (*sissors*) and *exhibition* (*exibition*). In fact, while forty-six of the forty-eight misspellings of *exhibition* began *ex*, only two began *exh.* Another word I could have chosen from the same test was *mortgage.* Out of seventy-three misspellings of *mortgage,* only eight contained a *t* (seven *mortage* and one *morgate*); the single most common misspelling was *morgage,* written by thirty-four of the students.

Double letters, or single letters that could be double, are also hard spots. The problems with *successful* clearly centre round the *cc*, the *ss* and *l.* Another word in the test that caused similar trouble was *disappoint;* out of seventy-four misspellings of *disappoint*, only two had *pp.*

A hard spot also occurs when there are several plausible renderings of a particular phoneme, especially if the correct one is not the most obvious. *Politician* and *ecstasy* were also in the test. Thirty five students ended *politician* with *tion.* Out of eighty-eight misspellings of *ecstasy,* all but eleven began with *ex.* This problem of having too many possible renderings to choose from is particularly acute with unstressed vowels. The *i* of *ventilated* was written *e, a, o, u, y, er, or* and *ur.* Out of eighty-seven misspellings of *definite,* only three had the second *i.*

The misspelling *definate* is a good illustration of the inflexibility of spelling in contrast to other aspects of language. In the areas of usage or pronunciation, if the majority of people start using or saying a word in the 'wrong' way, then what used to be 'wrong' becomes 'right'. If most people use *criteria* as a singular rather than a plural, then, in time, its use as a singular becomes standard. If most people pronounce *nephew* with a /f/, then that pronunciation becomes the standard one. But, with spelling, this does not happen. It appears from Table 5.2 that people who use the spelling *definate* certainly outnumber the users of *definite,* perhaps by as many as two to one, but *definate* does not thereby become correct.[2]

These hard spots – silent letters, double letters, unstressed vowels and so on – are places where the pronunciation does not give good guidance as to the spelling. This suggests that people are guided (or misguided) by the pronunciation of a word when they are trying to spell it, and there is other evidence of this.

In the last chapter I described a large corpus of errors taken from free writing and I noted that about half of the errors were homophones of the correct words, i.e. they would be pronounced in the same way, such as *fersilaty's* for *facilities.* One reason why the misspellings preserved this relationship to the pronunciation of the correct words is presumably that the writers had the pronunciation in mind when they wrote them.

People's accents sometimes affect their spelling. Rural people in West Virginia pronounce /ɪ/ as /i:/, e.g. *still* as *steel.* A spelling test (Boiarsky 1969) found that their spelling errors followed suit: a large minority of them misspelt *still* as *steel* or *steal,* whereas this particular error was not made at all by a group of students, similar in other respects, in Philadelphia. Another study (Graham and Rudorf 1970) compared schoolchildren in Georgia, Massachusetts and Ohio and found that the Georgians did better on *wh* words (*where, whale, whirl* and so on), because these words are pronounced with a /hw/ in Georgia.

One dialect which has had more attention than most is the dialect spoken by many black Americans, sometimes called 'Black English'. One feature of this dialect is the tendency to omit the final consonant in a word ending with two consonants, so that, for example, *planned* and *missed* would be pronounced like *plan* and *miss.* The research results are not clear cut but there is some evidence that schoolchildren who speak Black English are more likely to make corresponding errors in their writing, producing *plan* for *planned* and *miss* for *missed* (Desberg et al. 1980).

The influence of pronunciation can be seen also in the errors made by non-native speakers of English. Singaporeans, for example, often conflate /θ/ with /t/ and /ð/ with /d/ when these phonemes occur at the beginning of syllables, thus making homophones of *team* and *theme, den* and *then,* and this was reflected in a corpus of misspellings taken from the classroom essays of fifteen-year-old Singaporean schoolchildren (Brown 1986). Examples include *tin* for *thin, taught* for *thought* (and vice-versa),

bordering for *bothering* and *lather* for *ladder.*

More subtle evidence of the influence of pronunciation on spelling comes from studies of people with particular spelling problems. One small group of people consistently have trouble spelling words with consonant clusters that contain /l/, such as *split;* the *l* is either omitted or misplaced (Marcel 1980). Their hearing and their pronunciation appear to be normal, but, when they are asked to analyse the sound structure of one of these clusters, they seem unable to identify the /l/ as a separate phoneme. Consequently they have no reason, on the basis of their analysis of the pronunciation, to expect an *l* in the spelling, so they make the same mistakes with the *l* that people in general make with silent letters. They have the same trouble with *r.*

In fact poor spellers generally are not good at analysing the pronunciation of a word into its constituent phonemes (Baron et al. 1980, Perin 1983). Presumably this partly accounts for their poor spelling. However, it also works the other way. When good spellers do a phonemic analysis, they use their knowledge of the spelling to guide them, and this generally helps them to get it right (though it can also sometimes mislead them). Poor spellers do not have this assistance, so they find it harder.

What I have just said implies that there is always just one correct phonemic analysis of a word, but things are arguably more complex than that. Consider words like *can't* and *bent.* The standard analysis of *can't* is into four phonemes /kɑ:nt/, but it could plausibly be analysed into three, especially when pronounced by an American – /k/, a nasalized /ɑ:/ and a /t/ – and there is evidence (Read 1973) that some young children do analyse the sound in that way. This may explain why they sometimes misspell *can't* as *cat.* There is no special letter for representing a nasalized /ɑ:/, so they use the nearest one they can find – *a.* That adults regard this three-phoneme analysis as wrong may be at least partly because they are influenced by their knowledge of the spelling; knowing that there is an *n* in the spelling, they look for, and think they find, a /n/ in the pronunciation (Skousen 1982). On the one hand, then, a phonemic analysis can guide (or misguide) spelling; on the other, a knowledge of the spelling can guide phonemic analysis in helping the person to choose between different plausible analyses.

All that I have said so far supports the view that people often use their knowledge of the pronunciation of a word when they are trying to spell it. They make mistakes because they have difficulty analysing the pronunciation, or because the pronunciation – whether for all speakers of English or for them in particular – is a poor guide. However, not all misspellings come about because of pronunciation. Half of the errors that were analysed in the last chapter were homophones of the correct words, but half were not.

Word frequency is an important factor. An unfamiliar word is more likely to be misspelled even though its spelling might be close to the pronunciation. An American psychologist constructed a spelling test with words in the following four categories (there were ten words in each category – I give just three examples of each) and administered it to forty-seven students at UCLA (Brown 1970):

	High frequency	*Low frequency*
'Regular spelling'	discharge	calumny
	neglect	palimpsest
	visitor	sinuous
'Irregular spelling'	deceive	ocelot
	handkerchief	phosgene
	laughter	rhizome

A 'regular spelling' here means a spelling that is readily predictable from the pronunciation. Not surprisingly, the *discharge* group caused the least trouble (only one per cent of the spellings were incorrect) and the *ocelot* group caused the most (seventy-three per cent wrong). More interestingly, the *deceive* group were spelt much better than the *calumny* group (six per cent wrong for the *deceive* words, but forty-two per cent for the *calumny* ones).

It is obvious from all I said in Chapters Two and Three that you cannot spell English words solely on the basis of pronunciation. In a famous project in Stanford, California (Hanna et al. 1966), researchers catalogued all the sound-to-spelling correspondences in several thousand words and then wrote a computer program with over two hundred rules relating pronunciation to spelling. Each rule told the computer which spelling of a given phoneme was the most common, given that it was at the beginning or end of a word

or syllable or somewhere in the middle, and given that the syllable was stressed or not. The rules for /eɪ/ (the *a* of *spade*) were as follows:

	If it occurs here:	*Use this spelling:*
1	Initial	a .. e*
2	Medial unstressed	a .. e
3	Medial stressed followed by /l, m, n/	ai
4	Medial stressed	a .. e
5	Word-final	ay
6	Syllable-final	a

**a + consonant-letter(s) + 'silent' e*

The program tried the rules in the order given; for /treɪl/, for example, it would produce *trail* by rule 3 rather than *trale* by rule 4.

They then gave the computer the pronunciations of about seventeen thousand words, typed out in a standard form, and got it to generate spellings on the basis of its rules. It produced the correct spellings for about half the words. Table 5.3 gives some of its successes and failures.

Table 5.3 Spellings from the Stanford algorithm

Successes	*Failures*	*Should be*
abash	abaance	abeyance
abate	aflaim	aflame
abatement	afrade	afraid
abdicate	ale	ail
abdication	alwase	always
abdomen	baseting	basting
abolitionist	cafay	cafe
abrasion	colum	column
absorption	disapoint	disappoint
achievement	efitionsy	efficiency
actuality	fasanate	fascinate
adequate	iland	island
adherence	prosegure	procedure
advantageous	sicology	psychology
adventitious	spesamen	specimen

This result could be taken as showing how useless pronunciation is as a guide to English spelling – half marks is not a good score – but this would be too hasty. The program was restricted to choosing a spelling for each phoneme in isolation, knowing nothing about the context except its position in the word or syllable. It was not allowed to make any adjustments to its spelling of one phoneme in the light of its choices for neighbouring ones, so it spelled, for example, *belated* as *belateed (be-late-ed),* and many other words likewise. A lot of the program's errors were only slightly wrong, such as having a single letter instead of a double. To emphasize that it got half the words wrong seems a bit harsh. On the other hand, as a recent critique of this study has pointed out (Carney 1994), the pronunciation strings which constituted the program's input used a representation for unstressed vowels which gave the program extra clues about the spelling which are not present in ordinary pronunciation, enabling it, for instance, to spell correctly the final syllables of *avoidance* and *dependence*. Without these extra clues it would have made a lot more errors. Perhaps all the study shows is that pronunciation can give some help but that a simple-minded use of pronunciation, on its own, is not an adequate basis for good spelling.

With a large number of English words, including many of the commonest ones, people have to supplement their knowledge of the pronunciation with some information specifically about the spelling. Looking again at the misspellings of *scissors* in Table 5.1 on page 55, few of the students were trying to spell the word purely on the basis of its pronunciation. The most obvious sound-to-spelling rendering of *scissors* would be something like *sizzerz,* and none of the students produced this (though they did produce *sizers* and *sizzors*). Pronunciation was certainly one factor – the most common misspelling was one that simply omitted the 'silent' *c* – but the students were clearly also using their knowledge of how the word should look; they made errors because their knowledge was incomplete. The most obvious evidence of this is the appearance of *c*'s in odd places ('I know there's a *c* in it somewhere'). Most of them chose *o* for the second vowel, though there is nothing in the pronunciation to favour this choice. Many of them included a double consonant letter in the middle, though not necessarily *ss.* Not one of them ended the word with a *z.*

In order to spell oddly spelt words, people have to remember what's odd about them. Most of these students knew that the spelling of *scissors* was odd, and they had some idea of what was odd about it, but they did not know exactly, hence the variety of their mistakes. That English orthography contains so many odd features can cause people to make errors even when the correct spelling is straightforward. Misled by other, somewhat similar words that have odd spellings, people think, wrongly, that the word they are trying to write must have an odd spelling too. Examples of this are *wrotten* for *rotten, trambs* for *trams* and *gymn* for *gym.* The past tense of *lead* (*led*) is very often misspelt *lead,* presumably because the past tense of *read* (/ri:d/) is spelt *read* (/red/) and because there exists a word *lead* (the metal) pronounced /led/.

Another cause of misspellings that is not necessarily related to pronunciation is the incorrect division of a word into meaningful parts. The popularity of *sacreligious* as a misspelling of *sacrilegious* must surely arise from the temptation to divide it into *sac* + *religious.* It actually has nothing to do with *religious;* it is about stealing sacred things and the correct division is *sacri* + *legious.* Another example is *consensus.* People think there's a link with *census* and so produce the common misspelling *concensus.* They'd probably get it right if they saw the link with *consent.* The current popularity of *mini* as a prefix encourages people to write *minuscule* as *miniscule.* (It may be argued that both *sacreligious* and *miniscule* are errors caused by the pronunciation, but I suspect that the misspellings came first; people think they are saying *sacreligious* and *miniscule* and pronounce the words accordingly – an example, I suppose, of a phenomenon that would have to be called 'misspelling-pronunciation'.) *Highdraulic, corridoor* and *marballs* are further examples; hydraulic machines are often used for lifting things up high, corridors usually have doors leading off them, and marbles are little balls.

Many words in English are constructed by adding a suffix to a root word, and misspellings of these words suggest that people construct their spellings in exactly this way – by adding a suffix to a root (Sterling 1983). The simplest cases are where the suffix is just tacked onto the end, as in *packing* from *pack,* and *clocks* from *clock,* and these don't cause much trouble. But sometimes a small adjustment is required to the root and these are often misspelt.

Table 5.4 Some misspellings of inflected forms

comeing	19	diging	18	dinning	20
comming	5	bigging	2	dineing	2
cuming	2	biging	2	dyning	2
cameing	1	diding	2	bineing	1
cming	1	ding	2	danning	1
come	1	daning	1	dieing	1
comieg	1	degging	1	ding	1
comin	1	diggin	1	dinnie	1
cumin	1			dionig	1
going	1			doning	1
goming	1				
Correct	44%		26%		23%

babys	13	noticable	10	noticeble	1
babyes	4	notisable	6	notiesable	1
babyies	2	notesable	2	notiesball	1
bady	2	notticable	2	notisabl	1
badys	2	nessbell	1	notisbolle	1
abays	1	nocithisbord	1	notisbool	1
babbes	1	nolticable	1	notisdall	1
babe	1	norticabile	1	notisuble	1
babes	1	nosisboll	1	nowtisbull	1
babeyes	1	nosorbory	1		
babis	1	notabil	1		
babs	1	notcbell	1		
baby	1	noteisble	1		
baddys	1	notesably	1		
badies	1	noteseabol	1		
baybes	1	noteselb	1		
baybis	1	notetisable	1		
bayds	1	nothpes	1		
bebys	1	notible	1		
bobs	1	noticalbe	1		
dady	1	noticble	1		
Correct	17%				8%

The percentages are calculated out of all 730 students who took the test.
The number next to each misspelling is the number of students who produced that misspelling, out of a random sample of 83 scripts.

Table 5.4 presents some misspellings of *coming, digging, dining, babies* and *noticeable* taken from a spelling test given in the form of a dictation to a national sample of students on adult literacy schemes.[3]

The error is generally the failure to make the adjustment (*comeing, diging, babys*), but people sometimes make an adjustment when they shouldn't, as in *noticable,* or the wrong adjustment, as in *dinning* for *dining.*

A related type of error occurs in words made up of other words; people split the word into parts, as in *in side, other wise* and *any one.* Occasionally part of the result is not a word at all (*to gether*), and sometimes people respell some of the parts, as in *all ways* and *head miss dress* (for *headmistress*).

Since pronunciation obviously plays a large part in spelling, it is tempting to imagine that a writer begins with the pronunciation of a word in his head and converts this into writing using some sound-to-letter rules. The Stanford program, however, demonstrates that someone who did that would be a pretty bad speller. It is possible that a poor speller could spell like that, though there are plenty of other ways of producing misspellings, but it cannot be an accurate picture of what a good speller does.

A more refined version of this theory (Simon and Simon 1973), which was also implemented as a computer program, uses the sound-to-letter rules to generate not just one spelling of a word but a range of possible spellings. Each version is inspected, as it were, by the part of the brain which is used in reading. If the reading part decides that this version doesn't look right, the sound-to-letter system generates another version, and it carries on doing so until the reading part accepts one. For example, suppose we are trying to write the word *psalm.* The generator probably begins with *sarm,* that being the most obvious sound-to-letter rendering. The reading part rejects it. The generator tries other things for the *arm* part, among them *alm* (as in *palm, calm* and so on). The reading part prefers this. The generator also tries various other possibilities for the initial consonant, eventually coming up with *ps* which is rare but possible, especially at the beginning of a word.

The theory was designed to explain the way in which children produce spellings in spelling tests rather than the way in which competent writers spell familiar words. It explains spelling errors

by assuming that children vary in the number and complexity of their sound-to-letter rules and in the amount of information stored in the reading part. A child whose generator did not contain *alm* as one of its possibilities for /ɑ:m/ would be unlikely to come up with the right version. A child whose reading part knew nothing about the *p* or the *l* might write *sarm*; one who knew about the *p* but not the *l* might write *psarm* or *psam.* The program performs well in simulating this type of error.

It does not cope well, however, with exceptional spellings. The problem comes from the completely passive role of the reading part; it says Yes or No but it cannot make suggestions. Take *colonel* as an example. The reading part might know that there has to be an *l* in it, but it has to wait for the generator to come up with this in one of its offerings. In order for the generator to offer *colonel,* it has to have *olo* as a possible representation for the vowel /ɜ:/ in its table of rules, but this seems implausible. Whereas I might say that the misspelling *psociology* was strange but not unmotivated – a misapplication of the minor correspondence of *ps* to /s/ – I would say that *holomit* and *tolotle* for *hermit* and *turtle* were simply bizarre. There is no rule saying that *olo* might correspond to /ɜ:/. It's just a peculiarity of the word *colonel.*

However, whether or not this theory describes what people do when struggling to spell words in spelling tests, it does not describe what people do in the normal course of writing. Evidence on this comes from studies of people with acquired dysgraphia. 'Dysgraphia' means some impairment of writing ability, and 'acquired' means that it is due to brain damage, possibly caused by a head injury or a stroke. In one particularly clear case, a stroke patient had lost the ability to produce plausible spellings for non-words (such as *trid* or *fipe*) but could still spell correctly almost all the words he had been able to spell before his stroke. He could hear the non-words and could repeat them back. He could also read aloud a fair proportion of non-words that were presented to him written down. So, his ability to generate spellings from pronunciation had gone, but his spelling of known words was largely unimpaired.[4]

The inference drawn from cases such as this is that the spellings of words – all words, not just hard ones – are stored in a mental dictionary (or 'lexicon' as psychologists prefer to call it) and that

people begin the process of spelling a word by retrieving whatever information they have about the spelling in their mental lexicon. If this is true, there could be various ways in which a misspelling might be produced. The spelling in the mental lexicon might be wrong. Or it might be incomplete. Or the spelling of some other word might be retrieved. Or there might be a hiccup between the retrieval of the spelling and the actual writing or typing.

It is obvious that spellings in the mental lexicon are sometimes wrong. People can have arguments about how a word is spelt, and even place bets on it. Someone who thinks that *minuscule* is spelt *miniscule* will be surprised when the error is pointed out and may well not believe it until a dictionary is produced.

It seems equally obvious that spellings in the mental lexicon can be incomplete (or, if the complete information is there, it's hard to get at); presumably the enormous sales of dictionaries derive mainly from this fact. If a dictionary is not to hand, or if you cannot find the required word in it (a poor speller would not be able to find *gnome, pneumonia* and the like), how can you produce a spelling for a word whose spelling you are unsure of?

Generating a spelling from the pronunciation is certainly one possibility. Since English orthography is basically alphabetic, people can make some attempt to spell a word even if they have never seen it written down. They have to analyse the pronunciation into phonemes and then render each of the phonemes in some plausible pattern of letters. People vary in their idea of what the pronunciation is and in their ability to analyse it into phonemes.

They vary also in the sophistication of their knowledge of English orthography in general, and therefore in their notions of what patterns are plausible. A better speller would consider the possible effect of *e* or *i* on a preceding *c* or *g*, for example, whereas poor spellers tend not to take account of this. It is noteworthy, however, that even the worst spellers have some idea of what is acceptable. It is very rare, for example, for someone to begin a misspelling with a double consonant, and most people would know that they should not end a word with a *v*, even though the pronunciation might end with /v/, since this is also not permissible, except for some exotic or colloquial words. Even the invented spellings of very young children show some respect for

these patterns; they might use *ck* in a misspelling, as in *kack* for *cake,* but they rarely produce spellings like *ckak* – they know it can't come at the beginning (Treiman 1994).

Another approach is to spell by analogy with a known word. If asked to spell the slightly unusual word *leaven,* people might guess that it was spelt like *heaven.* (Or they might be unlucky and guess that it was spelt like *seven.*)

The experimental evidence is that people use a combination of the two methods. If they do not know the spelling of a word, they construct one using sound-to-letter patterns but their choice of patterns is influenced by known words that have some resemblance to the one they are trying to spell (Campbell 1983, Barry and Seymour 1988). When people are asked to write a non-word (which, by definition, they cannot retrieve from the mental lexicon), they have an idea of the various possible spelling patterns that can represent the phonemes in the non-word and they are guided in their choice by the frequency with which each pattern occurs in the language in general. However, their choice can be influenced by words they have just been looking at. If they are asked to produce a spelling for /zeɪl/, for example, they are likely to produce *zail* if they have just been looking at *snail,* but *zale* if they have just been looking at *stale.* Even a less frequent spelling pattern is susceptible to this sort of priming. If asked to spell /pi:m/, a few people will write *peme* if they have just been looking at *theme,* in preference to the more obvious *peam* or *peem.* A word that has been brought to mind without being actually presented can also have this priming effect; people who have recently heard the word *coffee* tend to spell the non-word /sti:/ as *stea* whereas people who have recently heard the word *forest* tend to spell it *stee* (Seymour and Dargie 1990).

If the word is an inflected form, they can spell the stem and then attach the inflection to it, perhaps failing to make the join correctly. If the word is composed of other words or of familiar word-fragments, the parts are spelled separately and strung together (or incorrectly kept apart). A mistaken analysis will tend to produce a misspelling, and even a correct analysis may sometimes do so, as in *proceedure* for *procedure* and *pronounciation* for *pronunciation.* Sophisticated spellers may guess at the origins of a word. If asked to spell /kɪ'rɒskəpɪ/ (a non-existent word), I would offer *chiroscopy,*

because it seems to belong with *chiropody, spectroscopy* and other such words derived from ancient Greek, though this too can have its dangers – people often misspell *crystallography* as *chrystal-lography,* thinking, perhaps, that the *ch* makes it look more academic.

With some words, pronunciation and word-analysis are not enough to give the correct spelling, and people just have to remember how to spell the hard parts. They may consciously employ rules-of-thumb ('*i* before *e* except after *c*' and the like) or particular mnemonic tricks for particular words. I was taught at primary school to remember that *height* is spelt like *weight.* I remember the difference between *practise* (verb) and *practice* (noun) by thinking of *advise/advice.* People sometimes remember a spelling-pronunciation, along with the correct pronunciation, such as /skɪzəz/ for *scissors,* /sepɑːreɪt/ for *separate* and /merɪŋgjuː/ for *meringue* (Ehri 1980). I spell *harass* by thinking of the American pronunciation with the stress on the *rass.*

To return to the psychologist's picture, a writer looks up a word in the mental lexicon and perhaps finds a spelling that is incorrect or incomplete. Another possibility is to find more than one spelling and not to know which is the one required. Homophones are the main problem here. Even though the correct spellings may be stored in the mental lexicon, there is a danger of choosing the wrong one. This is not too great a problem if the words are structurally different, such as *tide/tied* or *heel/he'll,* or if one is much more frequent than the other, such as *taught/tort,* but if the words do not have these distinguishing features, they cause great trouble. Poor spellers have difficulty with *there/their* and even good spellers trip up over *principle/principal* and *dependent/dependant.*

Even if the correct spelling is retrieved from the lexicon, there is a further danger of it being changed into something else before it gets written or typed. This is the topic of the next chapter.

I have said little so far about vision. There is a sense in which a spelling can 'look' right – I say more about this later. People sometimes say they have a 'photographic' memory for words and that, when asked to spell a word, they just 'see' it in their mind. But the results of an experiment suggest that a facility for visualizing words is not part of spelling ability, certainly not a necessary part. In this experiment, people had words read out to

them and were asked to say, for each one, how many letters it contained. Visualization was useful for this task – the people who visualized the words did better than those who did not – but this facility for visualizing was unconnected to their spelling ability. Some good spellers used visualization but other good spellers did not; some of the poor spellers also used visualization – presumably they often visualized wrong spellings (Sloboda 1980).

Another experiment casts doubt on the idea that the basis of good spelling is a facility for taking mental photographs. A group of good spellers and a group of poor spellers carried out the same string-comparison task. They were presented with pairs of items such as *splendid* and *splnedid* (i.e. words or small variations on words), some of which differed slightly in the middle and some of which didn't, and they had to decide for each pair, as quickly as they could, whether the two items were the same or different. The good spellers were quicker and made fewer mistakes. However, when presented with pairs of items made up of meaningless strings of consonant-letters such as *cdjpfslv* and *cdjfpslv,* or strings of non-alphabetic characters such as {&!#*<% and {&!*#<%, there was no difference between the two groups (Holmes and Ng 1993). The good spellers were not any better at processing arbitrary strings of symbols in an automatic, camera-like way; their superiority was confined to linguistically meaningful items.

The appearance of a word can vary enormously depending on whether it is handwritten, typed or printed or whether it is in upper or lower case. Most people could spot the misspelling in `ElEfAnT`, though they are unlikely to have seen the word written this way before. The mental lexicon seems to contain some sort of abstract specifications of letter strings rather than pictures of printed words.[5]

Reading and spelling are clearly not the same skill, even though we learn them at about the same time and though people who are good at one are generally good at the other. We all know words that we can read but can't spell. For some people, the disparity in attainment between the two skills is extreme. They have difficulty in segmenting words into phonemes and perceiving the relationships between sound and spelling – a characteristic of developmental dyslexia; they compensate by developing a reading strategy that relies heavily on the appearance of words and on context and

they achieve an acceptable level of competence at reading, but their spelling remains very poor (Frith 1980, Burden 1992).

The asymmetry of reading and spelling arises partly from a peculiarity of English orthography and partly from the nature of the task. Although the relationships between sound and spelling in English are ambiguous in both directions – an example would be the readings of *ea* on the one hand (including *treat, threat, great*) and the spellings of /i:/ on the other (including *green, clean, scene*) – the ambiguities are more numerous and more serious in the direction of sound to spelling.[6] In that sense, spelling is harder than reading. It is harder also because, to read a word, you need only extract enough information from it (and its context) to decide what word it is. You do not need to attend to all the letters; in fact you can probably manage even if some of the letters are rwong or m*s**ng. You don't have to know how to spell a word in order to read it.

In view of this it comes as a surprise to discover that young children who are in the process of learning to read and write can sometimes spell words which they cannot read (Bryant and Bradley 1980). It appears that, in the early stages, they use different strategies for reading and writing. They read by a word-recognition system but spell by a sound-to-spelling system, so they can read some words that they cannot spell, such as *school, light* and *train,* but they can also spell some words, such as *fit, cot* and *sunlit,* which they do not recognize as familiar written words and which they therefore cannot read.

A prominent theory of the development of literacy skills is based on the interaction between reading and spelling (Frith 1986). Early reading, it is suggested, proceeds by whole word recognition. When children begin to write, however, they adopt a sound-to-spelling approach. When they have established this for writing, they begin to use it also for reading, and their reading moves forward since they now have a way of attacking words they haven't read before. As their reading vocabulary increases, they acquire familiarity with the many non-alphabetic features of English orthography and they gradually incorporate these into their writing. Several studies of young children have lent empirical support to the earlier stages of this theory,[7] and a study of college students lends support to the later stage in showing that students who do more reading are better spellers, even after controlling for

other aspects of linguistic ability and intelligence (Stanovich and West 1989). More recent work puts less emphasis on the notion of stages and suggests rather that children from an early age have a mixture of skills which they bring to bear on the problems of reading and writing but that the prominence of this or that skill varies over time and, perhaps, from one child to another.[8]

Studies of older children and adults have also cast light on the relationships between reading and spelling. In one experiment (Campbell 1987), two students who were poor spellers were presented with lists of words containing some of their own habitual misspellings, the same words correctly spelt and some other correct and incorrect spellings. (They did not get correct and incorrect versions of the same word in the same list.) They had to say, for each word, whether the spelling was correct or not. While they successfully identified the correct spellings as correct almost all the time, they said 'Correct' to about half of their own habitual misspellings.[9] Two conclusions can be drawn. The first is that, if someone habitually makes the same misspelling – say *exturnal* – we cannot conclude that he thinks *exturnal* is right and *external* is wrong; given a test of this kind, he might say that *external* is right and *exturnal* is wrong. The second is that, just because someone can say that a spelling – say *refusal* – is right, we cannot conclude that he can spell it or even that, on another day, he will say that *rifusal* is wrong.

This second conclusion is supported by a study of two children, aged ten and twelve (Funnell 1992). In order for them to be able to recognize a correct spelling as correct, it was necessary only for the word to be one that they knew how to read. But in order for them to recognize a misspelling as incorrect, it had to be a word that they knew how to spell. For example, if they could read the word *antique,* they would recognize *antique* as correct but they might also say that *anticque* was correct. In order to recognize *anticque* as a misspelling, they had to be able to spell *antique,* not just read it. It follows from this that there is little point in telling poor spellers to check their work for spelling mistakes; they will fail to spot the mistakes for the same reason that they made them in the first place.

Being familiar with a word as a reader, then, is no guarantee of being able to spell it. On the other hand, when in doubt about a spelling, people sometimes write two possible spellings and then

choose between them; a spelling can 'look' right (Tenney 1980). Though reading requires less information than spelling, the reading process can be responsive to subtle features of a written word, and so people can use their reading skill to help with their spelling, as the following story illustrates. The word *format* can be used as a verb in computer parlance ('to format the output'), and a colleague asked me whether the *ing* form should be spelt with one *t* or two. I wrote them both down:

1. formating
2. formatting

You can see the problem. Version One looks as though it comes from a non-existent verb *formate* (like *refuting*), but Version Two is also unsatisfactory because the double *t* suggests that the stress is on the second syllable (like *rebutting*), whereas it is actually on the first (like *crediting*). Neither spelling was completely satisfactory, which was why he'd asked my opinion. We plumped for Version Two. The point of this story is that it was only by using our reading skills that we were able to make this analysis. We had to actually write them down in order to make a judgement. (At least one publisher, I have since noticed, has plumped for the other.)

It is only occasionally that people go to the trouble of writing down alternative spellings. A more usual use of reading when writing is in monitoring; people use their reading to check what they've just written. One psychologist (Sterling 1983) has suggested that this monitoring consists of a simple check that the word just written is an acceptable spelling, which might explain why the kind of error that consists of writing one word for another tends to slip through. People vary in their sensitivity to orthographic subtlety when reading, and this provides yet another source of variation in the type and number of spelling errors they are likely to make.

Marking the spellings in a spelling test as simply right or wrong is an injustice to the complexity of the process. People's efforts are not merely right or wrong; they approximate more or less closely to the correct spelling. Around each correct spelling there is, so to speak, a large family of potential misspellings related to it in complex ways, most of them closely enough to bear a family resemblance, others only distantly.

Notes

1. Table 5.1 presents some of the results from a spelling test given to 176 fifteen-year-olds in five comprehensive schools in inner London in 1980. The test was administered by Dr Dolores Perin as part of a screening procedure to identify subjects for psychological experiments concerned with spelling (Perin 1983). Table 5.2 presents a small part of the data collected by Dr Margaret Peters as part of her research on the teaching of spelling (Peters 1970). She administered a spelling test to the school-leavers in all the secondary schools in Cambridge in 1970. (This was part of the same exercise that produced the corpus analysed in Chapter Four.) This table contains the errors from a random sample of 110 of these students. The figures quoted in the next three paragraphs were taken from the same source as Table 5.2. In both the London and the Cambridge tests, each word was read out in a short sentence to give it some context. Both of these collections of errors, along with many others, are available in computer-readable form from the Oxford Text Archive (address on page 130) (Mitton 1985).

2. There are a few exceptions to this. *Wholism* seems to have become an acceptable alternative to *holism* (Upward, personal communication). *Donut* has become an alternative to *doughnut*. In an American telephone business index of 1974, fifty-eight businesses were listed under *doughnut*, all of which used *donut* in the name of the business or the product description (Jaquith 1976). In a way, however, this underlines the point I am making. These businesses were listed under *doughnut*, with an extra entry *donuts (see doughnuts)*, not vice-versa. Despite the unanimous preference for *donut* by the businesses themselves, the makers of the directory still felt that *doughnut* was the proper heading.

3. The National Foundation for Educational Research carried out a survey of 1,236 students in adult literacy schemes in England and Wales in 1978-79 (Gorman 1981). It included several tests, in increasing order of difficulty, and students could drop out of the testing if they found it too difficult. Seven hundred and thirty students took the test on which Table 5.4 is based. They were given a paragraph with blanks in it. Their tutor read out the paragraph, including the missing words, and they had to write the required words in the blanks. The percentages spelling the words correctly are calculated from the full sample. The misspellings are taken from a random sample of eighty-three of the scripts.

4. This patient was described originally by Shallice (1981). Patients have been reported with the opposite symptoms – they can produce plausible spellings

for non-words but they make many mistakes when writing real words, tending to bring the spelling into line with pronunciation. They seem to have lost the ability to retrieve spellings straight from the mental lexicon, but their sound-to-spelling generator is intact (Ellis 1984). There is a substantial literature on patients with impairments to this or that aspect of their literacy skills. See, for example, Morton (1980) or Allport and Funnell (1981).

5. There is some ambiguity in the word 'letter'. Are *h* and *H* the same letter or different letters? It depends on the context. If you say, '*Honesty* begins with an aitch,' you are talking about the eighth letter of the alphabet which can be written as *h* or *H;* in this sense they are the same letter. But if you say, '*Henry* begins with an *H* not an *h,*' you are talking about two different letters. It is for this reason that linguists sometimes use the term 'grapheme' (by analogy with 'phoneme') in preference to 'letter'. *H* and *h* are different realizations of the same grapheme (or, more technically, 'allographs' of the same grapheme). I have preferred to use the more familiar word 'letter' in this book since I have generally used it to mean the same as 'grapheme'. Besides, the term 'grapheme' is used in different ways. Some writers would call a digraph such as *th* a single grapheme; some even call *a . e* (as in *ape*) a single grapheme. Since we can spell words by writing in upper-case or lower-case or a combination of the two or by speaking the names of the letters or in various other ways, it appears that we store spellings in the mental lexicon at the graphemic level, though this begs the question of how we store information about capitalization – *March* (the month) is a different word from *march* (soldiers) and the *BBC* is not written *bbc.* I know of no experimental evidence on this.

6. I am here passing on an assertion made by Henderson and Chard (1980) and by Barry and Seymour (1988). It is based on analyses of the correspondences between phonemes and graphemes taken in isolation. The phoneme /f/, for example, can be written as *f, ff, ph* or *gh,* whereas the letter *f* is almost always pronounced /f/. Haas (1970), however, argues that these correspondences should be seen in the context of the words in which they occur. For example, the fact that /aʊ/ is written sometimes *ou (lout)* and sometimes *ow (howl)* might seem to present a problem for a writer, but if it is at the end of a word, it has to be *ow (how now).* If one assumes that a writer uses all the clues available both from the immediate context and from the language as a whole, he concludes that 'being burdened, and overburdened, with both kinds of divergent correspondence, [English] is just as troublesome to read as it is to write.' Carney also warns against drawing conclusions from decontextualized lists of grapheme-phoneme correspondences (Carney 1994).

7. For example, the studies reported in the collection edited by Sterling and Robson (1992), including Ellis and Cataldo (1992), Goulandris (1992), and Huxford et al. (1992).

8. See Goswami and Bryant (1990) and several chapters in the collection edited by Brown and Ellis (1994), including Ellis (1994), Treiman (1994), Lennox and Siegel (1994) and Snowling (1994).

9. Each student had two lists (so *external* would appear on one list, *exturnal* on the other). They worked through each list twice, with a week between one session and the next. Items to which they gave inconsistent responses ('Correct' one time and 'Incorrect' the other) were excluded from the analysis.

CHAPTER SIX

Slips and typos

This chapter deals with the sort of errors that occur when people know how to spell a word but inadvertently write or type something else. I will begin with slips of the pen and then go on to the kinds of error specifically associated with keyboard input.

If you spotted one of your own spelling mistakes, you would probably know whether it was just a slip, but it is not always possible to be sure about this if you are looking at someone else's work. Although some cases are clear enough – *the* for *then* is presumably a slip whereas *brtatons* for *potatoes* is not – how should one classify *campains, excercise* or *depleated*? The 'phonetic' nature of these errors might suggest that they are misspellings rather than slips, but this is not conclusive. Writers have reported catching themselves making phonetic slips; examples include *shure* for *sure* (Morton 1980) and *ridgid* for *rigid* (Hotopf 1980).

Some researchers have avoided this problem of trying to distinguish between slips and other sorts of error by collecting slips entirely from their own writing; whenever they have noticed themselves make a slip they have jotted it down. Ellis (1979) and Hotopf (1980) have made collections of this kind. The great advantage of this method is that they can be sure that the errors were indeed slips and they also know what words they were trying to write. A disadvantage is that all the slips in such a collection were made by just one person and it is possible that one person's slips differ from another's. Another way round the problem is to collect slips from passages in which the writers themselves have made corrections; the researcher notes the words or part-words that were written and crossed out. Hotopf has done this with

students' essays and Wing and Baddeley with examination scripts written by candidates for the entrance examinations to Cambridge colleges (Wing and Baddeley 1980). This method slightly widens the notion of a slip to include what one might prefer to regard as second thoughts. It is possible for someone to write, say, *depleated* and then to decide that it doesn't look right and to change it to *depleted,* which is not quite the same as saying that *depleated* was just a slip. The three examples given above – *campains*, *excercise* and *depleated* – are all taken from Wing and Baddeley's list of slips which were corrected by the writers and they look as though they could be of this type, though they could equally be genuine slips.

From the point of view of spellchecking, there is a serious drawback to these methods of collecting slips, namely that the errors were detected by the people who made them. A spellchecker is required to detect precisely those errors which the writers do *not* correct themselves. Errors of this kind – uncorrected slips – are harder to identify. A commonly adopted rule of thumb is that a mistake can be taken to be a slip if the writer spells the word correctly elsewhere in the text, but even this is not always conclusive. Suppose someone writes *their was* in one place and *there was* in another. The *their* might have been a slip, but it is possible that this person is unsure whether it is *their* or *there* that goes before *was* and sometimes uses one, sometimes the other. A further problem is that poor spellers often produce different misspellings of the same word and they might occasionally by happy chance produce the correct spelling. But I do not want to exaggerate these difficulties. Wing and Baddeley included uncorrected errors in their corpus, distinguishing between slips and other errors (which they call 'convention errors') by the above rule of thumb, and my impression, both from aspects of their analysis and from inspection of the lists of errors, is that their criterion was more likely to misclassify a slip as a convention error, just because the writer did not happen to use the word elsewhere, than to include convention errors as slips.

Handwriting slips

At the level of individual letters, slips are often caused by interference from other letters in the word. Sometimes a letter which is due to come later in the word or in the next word is brought forwards; examples from Ellis's collection of his own slips include *J.Seuro* when he meant to write *J.Neurol.Neurosurg.* and beginning the word *Cognitive* as *Go.* An interesting feature of these errors is that the wandering letter assumes the case (upper-case or lower-case) of the letter it displaces, as in both the above examples, which suggests that the displacement occurs before decisions have been made about the detailed form of the letter string which is about to be written.

Another kind of interference is the omission of a letter in the neighbourhood of another occurrence of the same letter, as in *satifactory* and *SHOR-TERM* (Ellis's examples again). The common *intial* for *initial* would be another example. In contrast to displacement errors, this kind of interference seems only to occur between letters written in exactly the same way; you don't get an upper-case letter causing the omission of a lower-case one – an example of this (invented, since they don't occur) would be *Georaphy.* Some people's handwriting contains two forms of lower-case *s* and, again, the occurrence of one of them does not cause the omission of the other. It appears that this kind of interference occurs between precisely specified letter forms.

A special kind of interference occurs with doubled letters; people are inclined to double a different letter in the same word. Ellis caught himself beginning *agrammatism* as *agrr.* An example from the Cambridge candidates is *Meditterranean* (from someone who spelt it correctly elsewhere). The same error has been noted in typing; examples include *diseect* for *dissect* and *scrren* for *screen* (Shaffer 1975, Norman and Rumelhart 1983). It is as if the letter string that is being produced contains a marker saying 'letter doubled' which is stored as a separate item from the specifications of the letters and which can occasionally get attached to the wrong letter. It is interesting that two computer simulations, the first of typing (Rumelhart and Norman 1982) and the second of spelling (Houghton et al. 1994), have found it necessary to make special

provision for doubled letters. The problem arises in these simulations because, after producing a letter, the system has to damp down the process that gave rise to that letter to prevent it from immediately producing the same letter again. This damping down, however, prevents it from ever producing a doubled letter, hence the need for something like a 'doubling' marker, separate from the letter it refers to.

In getting down the letters of a word in the right order, it is important to keep track of where you are. Some errors arise from failure to do so, in particular jumping forwards to a later occurrence of the letter you have just written, as in *depence* for *dependence* and *begas* for *began as.* The remarkably common *rember* for *remember* is another example. *Probly* and *libry* could also belong in this class, though pronunciation might also play a part with these.

A similar slip occurs at the level of the strokes that make up a letter. Ellis notes *REFECT* for *REFLECT, NAMNG* for *NAMING* and *langug* as the beginning of *language.* In *REFECT,* the strokes that make up the *L* are also the first two strokes for an *E*; having written the *L,* he carried on as though he was in the middle of the *E.* These errors are at the level of finger control rather than language processing; they occasionally result in the running together of two letters to make something which is not a letter at all, such as a capital *T* and an *h* where the vertical stroke of the *T* becomes the upright of the *h.* (Ellis apparently begins his *T* with the horizontal stroke.)

Other snippets of information from the literature on slips is that letters with ascenders (*b, d, f, h, k, l*) or descenders (*g, j, p, q*) are less likely to be omitted than other letters (Hotopf 1980), that the probability of making a slip increases steadily as you move through a sentence (Wing and Baddeley 1980) and that a slip is more likely to occur in the last word on a line (Smith 1983).

Moving from the level of letters to the level of words, the striking thing about slips is the proportion of real-word errors among them and the proportion of these real-word errors that are made up of function words. I commented on these in Chapter Four, and other writers (Sterling 1983, Hotopf 1980) have noted the frequency of slips of this type. Hotopf notes also that, where the error consisted of completely omitting a word, as opposed to writing something else instead, nine out of ten of the omitted

words were function words. I gave some tables of results on real-word errors from a corpus of compositions by school-leavers in Chapter Four. Table 6.1 presents some figures from the corpus of uncorrected slips made by the Cambridge candidates.

Table 6.1 Real-word slips

Total uncorrected slips	254	
Real-word errors	109	43% of 254
Error was a function word	41	38% of 109
Both error and target were function words	35	85% of 41

Whereas the school-leavers corpus included word-division errors (*to gether, be side* and the like), the Cambridge candidate corpus does not, so all the real-word errors in Table 6.1 consist of one word written for another. The 'target' is the word the writer intended to write.

Real-word errors and slips are not the same thing; not all real-word errors are slips and many slips result in non-words. But there is considerable overlap, and they are of special interest from the point of view of spellchecking since they cannot be detected by simple dictionary look-up. Some of them will be accidents – a writer might spell according to the sound and produce a word which just happens to be in the dictionary, such as *pict* for *picked* – but it is unlikely that this accounts for more than a small proportion of them. A sound-to-spelling system rarely generates dictionary words – the Stanford program that I mentioned in the last chapter generated spellings in precisely this way and it occasionally produced real-word errors, such as *ale* for *ail,* but only six per cent of its errors were of this kind. A high proportion of wrong-word errors are on short words that almost everyone knows how to spell, such as *her* and *than,* and over half the wrong-word errors in the school-leavers corpus were not homophones of the intended words. When homophone errors do occur, it is generally obvious that the writer has selected a known spelling rather than generated a plausible one. For example, if you constructed a spelling for *there* on the basis of sound-to-spelling rules, you would expect to produce such versions as *thair* and *thare,* but in fact the word that gets written is almost always *their.* It appears that the majority of wrong-word errors arise because the writer makes the

wrong choice from a pair of words that look or sound similar or intends to write one word but in fact produces another.

In short, real-word errors and non-word errors are different. This is underlined by the figures in Table 6.2, taken from the candidate corpus of uncorrected slips. To generate this table, I compared each error with its target, moving left to right through the word, and noted the first place at which they differed. This position was then assigned to one of five sections according to a system described by Wing and Baddeley. Section 1 is the early part of the word, 2 is left of centre, 3 is the middle, 4 is right of centre and 5 is the end part.[1] For example, the error *competion* first differs from the target *competition* at the eighth character position; according to the system, the eighth letter of an eleven-letter word is in section four. I have subdivided section 1 into 1A – the very first letter – and 1B – the rest of section 1.

Table 6.2 Position of first wrong letter

Section	*Real-word*	*Non-word*
1A	6%	2%
1B	0%	5%
2	1%	22%
3	7%	30%
4	16%	18%
5	70%	23%
n=100%	109	145

The most striking feature of Table 6.2 is that most of the real-word errors matched their target closely and then differed at the end.

Half of them were caused by the writers simply leaving the target words unfinished, as in *are* for *area*, *be* for *been* and *though* for *thought*. Many of the omitted fragments were inflections – *find* for *finding*, *reach* for *reached*, *person* for *persons*. By contrast it was rare for an uncompleted word to be a non-word error. Only four per cent of the non-word errors were uncompleted words; examples include *directio* and *structu*.

Another twenty-eight per cent of the real-word errors were identical to the target words up until the last letter of the error word, at which point they turned into a different word, as in *at* for

as, behave for *behaviour, notably* for *notable* and *word* for *world.*

Several psychologists (Ellis 1979, Smith 1983, Sterling 1983) have suggested that we read what we are writing as we go along to make sure that it is what we intended to write. Though the purpose of this monitoring is to catch errors, it can occasionally provoke them. Perhaps the eye sees the fingers complete a familiar word, such as *the* or *how* or *special,* and signals the hand to move on to the next word even though the intended word was *then* or *however* or *specialize.* Perhaps reading *defin* puts you in mind of the word *define,* though what you meant to write was *definable.*

Though the main difference between real-word and non-word errors shown in Table 6.2 is that the real-word errors tended to differ from the target towards the end of the word, there was a small group of real-word errors that did the opposite – they differed in the first letter (shown as subsection 1A). This difference in Table 6.2 is statistically significant only at the 0.1 level, but the finding is supported by the appearance of the same pattern in the school-leavers corpus where ten per cent of the real-word errors were wrong in the first letter compared with five per cent of the non-word errors. These are words that look or sound like the intended word; examples include *as* for *is, as* for *us, know* for *now, new* for *knew, right* for *write* and – a common one with the school leavers – *are* for *our.*

Though pronunciation plays a less obvious part in slips than in other sorts of misspelling, it still has an effect. I mentioned earlier the occurrence of phonetic slips such as *ridgid.* Sterling notes also that pronunciation has an effect with omitted inflections. The past tense inflection *ed* is less likely to be omitted if it is syllabic – *needed* or *wanted* as opposed to *considered* or *searched.* The candidate corpus and the school-leavers corpus support this. Inspection of five pages from books on my bookshelves suggests that about a fifth of past tense *ed* inflections in running text are syllabic. Of the nine errors in the candidate corpus which consisted simply of leaving off the *ed* or *d*, only one was syllabic. For the school-leavers corpus, the figure was three out of sixty-seven (not counting the sixty-six occurrences of *use* for *used.*)

Typing errors

Composing text with a typewriter has been the custom for professional authors and journalists for many years but the wordprocessor has made the keyboard commonplace. The keyboard provides writers with yet another way of making errors. Some errors in typewritten material have their counterparts in handwritten text – I referred above to the occasional doubling of the wrong letter, an error common to both. In this section, however, I concentrate on errors which are specific to keyboard input. Figure 6.1 shows the layout of the standard qwerty keyboard.

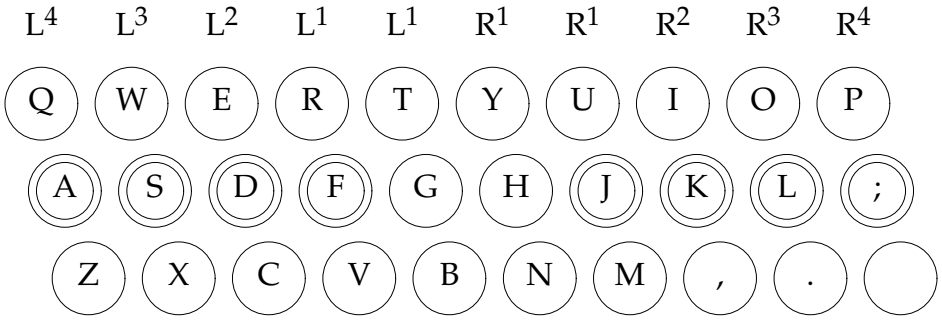


Figure 6.1 The Qwerty keyboard

The letters above the top row of keys indicate the fingers which are normally used to strike the keys in each column in the standard method of touch-typing. L^1, the first (index) finger of the left hand, deals with two columns, as does R^1. The keys with double circles are the 'home' keys, where the typist rests the fingers when they are not striking a key.

As is well known, when Christopher Latham Sholes designed the qwerty keyboard in the 1870s, the comfort of the typist was not uppermost in his mind (Cooper 1983). A problem with the early machines was that, when a key had struck the platen, it was slow in falling back to its place in the type basket and sometimes collided with a key that was on its way up. Minimizing this problem was the main motivation for the qwerty layout. Other keyboard layouts have been designed, notably the Dvorak keyboard in the

1930s. This is completely different from the qwerty layout; the left-right sequence on the home row, for example, is A O E U I D H T N S. A greater range of common words can be typed with the home row keys; the right hand is used more; the work is more evenly distributed among the fingers; more use is made of alternating hand sequences, which are faster, and there is less need for awkward finger movements. In extensive tests, it has been found that people can learn the Dvorak keyboard more quickly; they can type faster and with fewer errors and less fatigue. But we seem to be saddled with the qwerty one, though comparisons of various designs suggest that qwerty is not as bad as it is sometimes made out to be; it is certainly possible to design a far worse keyboard (Norman and Rumelhart 1983).

Typing has been of great interest to psychologists ever since it became a widely practised skill at about the turn of the century. In laboratory experiments, a person can react to some stimulus – say pressing this or that button in response to a light – in about a quarter of a second. On this basis you would predict that a copy-typist, looking at some text and reacting to each character by pressing a key on the keyboard, should be able to type about four characters per second, or roughly fifty words per minute. But in fact speeds of a hundred words per minute, and more, are not unusual (Salthouse 1984). So, if typists are not reacting to one character at a time, what are they doing? The current consensus is that they achieve these speeds by overlapping different operations; while one key is being struck, the fingers are arranging themselves for the next strokes while the eyes are taking in a few more characters. It resembles what a computer scientist would call 'pipelining'.

Evidence of this comes from slow-motion videotape of typists' fingers. The diagram on the next page is based on a videotape of a typist typing the word *pining*.[2] It shows the movement of a white dot stuck on the fingernail of the index finger of the typist's left hand.

Time in the diagram is going from bottom to top. It is measured in milliseconds, counting down to the striking of the *g*. The small circles show the position of the dot at intervals of twenty milliseconds. All the letters of *pining* except the last are typed with the right hand. While the *pi* is being typed, the finger waits on its

home key (the *F* key). When the first *n* is struck, it begins to move to the right towards the *G* key and it accelerates while the *i* of *ing* is being struck. By the time the right hand strikes the *n* of *ing*, it is poised above the *G* key ready to strike. This is characteristic of skilled typing; the fingers begin to move to their keys two or three letters in advance of their turn.

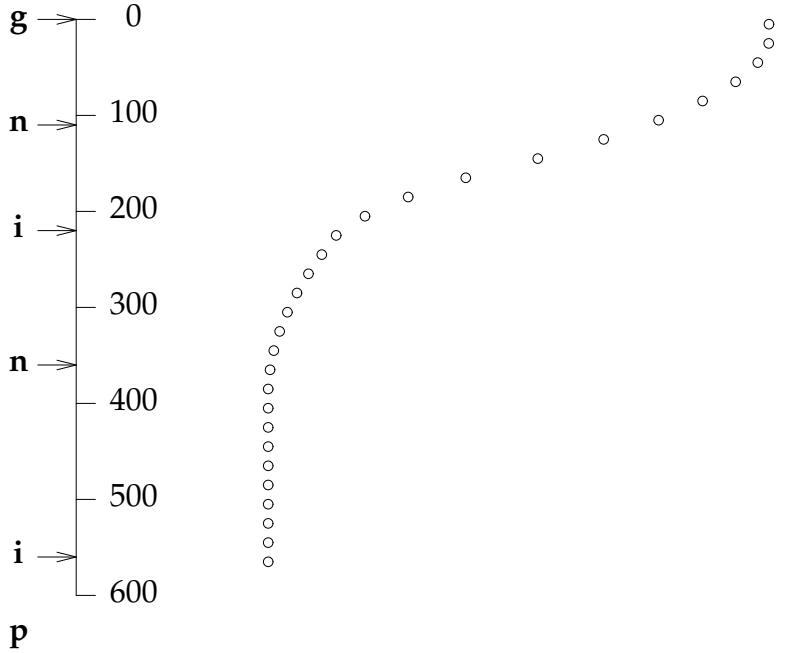


Figure 6.2 The movement of a typist's left index finger while typing 'pining'

Nearly all the research on typing has been done on copy-typing, where the typist is typing out a document already written. Compositional typing is presumably different but it is not clear exactly what the differences will be. Obviously, a writer is thinking about the meaning of the text whereas a copy-typist does not have to. It makes little difference to copy-typists whether they pay attention to the meaning of the text or not (Cooper 1983, Salthouse 1986). In fact a typist can type a text composed of random words as fast and as accurately as ordinary prose (Shaffer and Hardwick 1968).

Obviously, typing speed depends on the skill of the typist but, apart from that, it depends on the layout of the keyboard and on the text being typed. For a skilled typist, the time gap between one

keystroke and the next is shortest for keys typed with different hands (such as *sk*) and longest for keys typed with the same finger, such as *sw.* Keys typed with different fingers of the same hand, such as *sc,* are generally intermediate but this varies with the precise letters being typed and varies also from one typist to another. Some typists are able to move their fingers independently so that the movements for a two-finger digraph can overlap as for a two-hand one; when typing *in,* for example, the right index finger can get ready for the *n* while the second finger is striking the *i.* With other typists the movement of one finger interferes with that of another finger on the same hand, so that, in this example, the index finger's preparations for the *n* would be disrupted by the second finger's striking of the *i* (Gentner 1983).

Superimposed on these physical effects of the keyboard are the effects of digraph frequency in the language; more common digraphs are typed faster. That this is a genuine effect of language was demonstrated in an experiment which compared the typing speeds of Dutch and American typists. Certain digraphs are more frequent in English than in Dutch while others are more frequent in Dutch than in English; *ab* is an example of the first kind, *ba* of the second. The American typists were faster with the first kind, the Dutch typists with the second (Gentner et al. 1988). This seems to be the result of prolonged practice. There is also a more short-term effect of word frequency. Even within a single document, typists improve their speed for a word that occurs in it many times. For example, having typed this book myself, I can now rattle off the word *misspelling* with some panache (by my standards).

Typing errors are generally classified into the familiar four categories:

one letter omitted	*wrd*
one letter inserted	*woird*
one letter substituted for another	*woud*
two adjacent letters transposed	*wrod*

About nine out of ten mistyped words contain just one of these errors. Learners make a lot of substitution errors. This problem diminishes with increasing skill so that, for skilled typists, it is insertion errors that predominate (Grudin 1983). However, about four fifths of these errors are spotted by the typist (Long 1976).

Typists detect their errors remarkably fast. Even though they are bowling along at seven or eight letters per second, they often stop immediately they have made an error and they rarely type more than another two letters (Shaffer and Hardwick 1969). In fact they sometimes know they are about to make an error before they have made it. When a typist makes an error and then immediately stops, it often turns out that the stroke that was made in error was made more lightly than usual, as though the typist was trying to retract it (Rabbitt 1978).

Some kinds of error are harder to detect than others. Omissions, in particular, are likely to be missed (Shaffer 1975). Consequently, though omissions form a small proportion of the errors originally made, they form the largest group of uncorrected ones.

Studies of uncorrected typos face the same data-collection problem as studies of slips of the pen; it is easy enough to collect errors from keyboarded text, but it is impossible to separate the typos from the misspellings. A project known as SPEEDCOP (Spelling Error Detection and Correction Project), which was aimed at the automatic correction of errors in a large database of chemical abstracts, extracted a large corpus of non-word errors from over twenty-five million words of scientific text (Pollock and Zamora 1983). The breakdown of the SPEEDCOP errors into the four categories is shown in Table 6.3.

Table 6.3 Errors in a large corpus of keyboarded text

Non-word contained:		
a single omission error	34%	
a single insertion error	27%	
a single substitution error	19%	
a single transposition error	12.5%	
more than one error	7.5%	
Total non-words (=100%)		*52,963*

Looking at the errors in more detail, there is surprisingly little relationship between the patterns observed in studies of typing,[3] which collect all errors that typists make as they go along, whether they correct them or not, and the patterns in the SPEEDCOP errors, which are obviously uncorrected ones. I can think of two possible explanations for this. The first is that a typist's success at detecting

errors varies so much with the type of error that the profile of uncorrected errors is completely different from that of errors originally made, so that, in looking at uncorrected errors, we are looking not so much at the effects of those processes which give rise to errors in the first place as at the effectiveness of those processes by which typists detect them. The second is that, though the SPEEDCOP researchers were of the opinion that most of the errors in their corpus were typos, it is possible that a significant minority were spelling errors. In some ways the SPEEDCOP results do resemble those from a collection of spelling errors. These two explanations could be related; perhaps typos that give rise to plausible misspellings are less likely to be spotted.

Taking the error types one by one, many omissions are caused by the finger making only a half-hearted movement towards the key or not striking it with sufficient pressure (Shaffer 1975), though videotape analysis shows that, in about half the cases, there is no motion towards the key at all. Salthouse suggests that keys struck by the little fingers are more likely to be omitted.[4] I noted in my discussion of slips of the pen that letters which had recently occurred in a word were more likely to be omitted, as in *satifactory*. The same pattern has been observed in typing slips – an example is *artifical* – though Salthouse found no sign of this. More reliable are the findings that the first letter of a word is rarely omitted and that there is a tendency to omit one of a double-letter pair.

Insertion errors are predominantly misstrokes – one finger hitting two keys or occasionally a neighbouring finger coming down at the same time, so an inserted letter tends to be a keyboard neighbour of an adjacent letter in the text. These patterns can be idiosyncratic. Grudin describes one typist who tended to skim close to the *k* after hitting the space bar. When the next letter happened to be a *p,* the reaching for the *p* caused the skimming finger actually to strike the *k.* Consequently, her only insertions of the letter *k* were before words beginning with *p.*

A second form of insertion is the incorrect doubling of a letter. Salthouse reports that sixteen per cent of insertion errors were of this type, but they account for almost half the insertion errors in the SPEEDCOP corpus, which suggests that they are more likely to go undetected.

Substitutions are more interesting. Grudin presents figures from a corpus of over sixty thousand substitution errors. Figure 6.3 draws on that portion of the figures relating to the letter *d;* it shows the number of times each of the letters on the keyboard was typed in place of a *d.* The amount of shading in a box represents the number of times that that key was typed in place of *d.* For example, the *S* was the key typed most often in place of *D,* the *E* key next most often, then the *F* and *C* keys and so on.

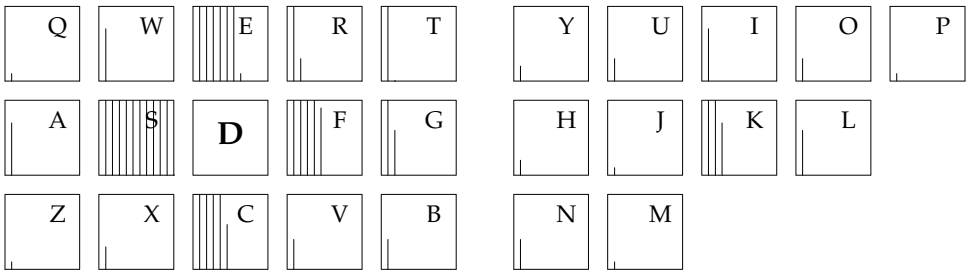


Figure 6.3 Keys typed in place of 'd'

The keys in Figure 6.3 are aligned to show how they lie beneath the typist's fingers. *E, D* and *C,* for example, are all struck by the same finger (the second finger of the left hand). The wider gap in the middle is to emphasize the separation of the keyboard between the two hands.

Seeing the high proportion of substitutions in which a keyboard neighbour is typed in place of the correct letter (*s, e, f* and *c* in Figure 6.3), it is tempting to assume that these are simply misstrokes, like insertion errors. But this is not so. In substitution errors, the key is struck by the finger that normally strikes it. These errors are not bad shots at the right keys; they are good shots at the wrong keys. The specification of a finger action has to select hand, finger and row (and, for index finger, it must also select the inner or outer position). If one of these parts of the specification is wrong, you get a substitution error. For *d,* the wrong finger gives *s* or *f* or, more remotely, *a,* the wrong row gives *e* or *c* and the wrong hand gives *k.* Getting two parts of the specification wrong is less likely, but a few of these can also be seen in the diagram.

Letter frequency also has an effect; a frequently occurring letter is likely to be substituted for a less frequent one. For example, *d* and *k* form a keyboard pair (same finger, same row, different hand), but *d* is more frequent than *k.* The proportion of *k's* that are replaced by *d's* is higher than the proportion of *d's* that are replaced by *k's.*

The SPEEDCOP corpus confirms the frequency of keyboard-neighbour substitution errors, but also shows a strikingly high proportion (forty per cent) of vowel-for-vowel substitutions. Substitutions of this kind do occur more often than you would expect just on the basis of letter frequency and keyboard layout, but, even so, the figure from the typing studies is only about fifteen per cent. Vowel-for-vowel substitutions are more characteristic of spelling errors than of typos; over half of the single-substitution errors in the school-leavers corpus (the one described in Chapter Four) were vowel for vowel. This makes me wonder whether a significant proportion of the SPEEDCOP errors were in fact spelling errors, or alternatively whether typos that produced plausible-looking misspellings were less likely to be detected by the typists.

Finally transpositions. The salient feature of these is that eighty per cent or more involve keys typed with different hands – an example would be typing *th* as *ht*.[5] Many of them occur in short function words. Because so many transposition errors involve alternate-hand sequences, the function words that are particularly affected are *the, that, than, for* and *to,* as opposed to function words typed with one hand, such as *are, as, at, be, in, on* and *was.* They also tend to affect common digraphs. Since common, two-hand digraphs are typed especially fast, it is tempting to suppose that the two fingers are coming down almost together and that the second finger just gets there first occasionally. But this would predict a very short time gap between one stroke and the next in transposition errors, and researchers have not consistently found this. Additionally, videotapes of transposition errors show that the second finger (such as the one that strikes the *h* of *th*) actually begins its movement towards its key before the other. It appears that transposition errors occur in the head rather than in the fingers.

Notes

1. Letter positions were assigned to sections as follows:

Words of length:	*broken into sections thus:*				
	1	2	3	4	5
1	-	-	1	-	-
2	1	-	-	-	2
3	1	-	2	-	3
4	1	2	-	3	4
5	1	2	3	4	5
6	1	2	3,4	5	6
7	1,2	3	4	5	6,7
8	1,2	3	4,5	6	7,8
9	1,2	3,4	5	6,7	8,9
10	1,2	3,4	5,6	7,8	9,10
11	1,2	3,4	5-7	8,9	10,11
12	1-3	4,5	6,7	8,9	10-12
13	1-3	4,5	6-8	9,10	11-13
14	1-3	4-6	7,8	9-11	12-14
15	1-3	4-6	7-9	10-12	13-15

   Errors that consisted of the complete target followed by superfluous letters, such as *usually* for *usual,* were assigned to section five.

2. This diagram is taken from a paper by McLeod and Hume (1994).

3. The rest of the discussion of typing errors draws mainly on Grudin (1983) and Salthouse (1986).

4. The omission data from the SPEEDCOP corpus do not have any obvious pattern at all and bear no relation to Salthouse's results. Taking the number of times a letter was omitted as a proportion of the number of times it occurred in the text (i.e. controlling for letter frequency), the letter most likely to be omitted in Salthouse's study was *v*, followed by *s*, *p*, *g*, *w*, *c* and *a*, in that order. In the SPEEDCOP data, it was *w*, followed by *e*, *r*, *i*, *s* and *a.*

5. Oddly, I found that seventy per cent of the transposition errors in the school leavers corpus were alternate-hand sequences. Since these pieces were entirely handwritten, this obviously had nothing to do with the qwerty keyboard. It seemed to be caused largely by *ei/ie* confusions.

CHAPTER SEVEN

Spelling checkers and correctors

By the standards of the computer industry, spelling correction has a long history; people have been writing programs to detect and correct spelling errors for over thirty years. Reviews of the literature are provided by Peterson (1980a, 1980b) and Kukich (1992a). In this chapter I sketch the main methods and some of the unsolved problems. I will simplify the descriptions so as not to get bogged down in the detail.

The most popular method of detecting errors in a text is simply to look up every word in a dictionary; any words that are not there are taken to be errors. But before I describe variations on this method, I will mention two that do not use a dictionary in this way.

The first uses a dictionary indirectly (Riseman and Hanson 1974). It begins by going right through the dictionary and tabulating all the trigrams (three-letter sequences) that occur; *abs*, for instance, will occur quite often (*absent*, *crabs*) whereas *pkx* won't occur at all. Armed with this table, the spelling checker divides up the text into trigrams and looks them up in the table; if it comes across a trigram that never occurred in the dictionary, the word that contains it must be a misspelling. It would detect *pkxie*, for example, which might have been mistyped for *pixie*. For detecting people's misspellings, this technique is of limited value since a high proportion of errors do not contain any impossible trigrams, but it is of some use in detecting errors in the output of an optical character reader (a machine that scans a page of text and 'reads' the letters).

The second does not use a dictionary at all (Morris and Cherry 1975). Like the previous method, it divides the text into trigrams,

but it creates a table of these, noting how often each one occurs in this particular piece of text. It then goes through the text again calculating an index of peculiarity for each word on the basis of the trigrams it contains. Given *pkxie,* for instance, it would probably find that this was the only word in the text containing *pkx* and *kxi* (and possibly *xie* too), so it would rate it highly peculiar. The word *fairy,* by contrast, would get a low rating since *fai, air* and *iry* probably all occur elsewhere, perhaps quite often, in the passage being analysed. Having completed its analysis, it draws the user's attention to any words with a high peculiarity index. Like the previous method, it would fail to spot a high proportion of ordinary spelling errors, but it is quite good at spotting typing errors, which is what it was designed for. An advantage that it has over all dictionary-based methods is that it is not tied to English; it will work on passages of, say, French, German or Greek.

The majority of spelling checkers, however, use a dictionary in some way. I say 'in some way' because they do not necessarily hold a complete dictionary with all the words spelt out in full, though some do. Some economize on storage space by holding only the stems of words (McIlroy 1982). For example, instead of holding *doubt, doubts, doubted* and *doubting,* they hold just *doubt* and use a set of rules to remove suffixes before looking words up; given *doubting,* the checker would remove the *ing* and look up the *doubt.* They may remove prefixes also (*undoubted*) and they may carry on removing suffixes (or prefixes) until they reach a stem (*undoubtedly*). The process is known as 'affix-stripping'.

The rules have to be a bit more complicated than this in order to cope with such forms as *cutting* (to get *cut* rather than *cutt*), and *denied* (to get *deny* rather than *deni*). The rules have to have some ordering, so as to accept *undoubtedly* but not *undoubtlyed,* and they need to have some way of coping with words that look like inflected forms but aren't, such as *farthing.* The strength of this system is that the checker can accept freshly minted words that are acceptable but are possibly not in any dictionary, such as *unplaceable.* The weakness is that it will accept some words that don't exist, such as *undoubt.*

A second way to save storage space is to hold the dictionary as a bit map (McIlroy 1982, Nix 1981). Imagine the memory of a computer as a long row of lightbulbs, initially all switched off. You

go through the dictionary and convert each word, somehow, into a number. For example, you might start by converting *a* to 1, *b* to 2, *c* to 3, and so on; the word *ace,* for example, would become 1,3,5. Then multiply the first number by 1, the second by 2 and so on, and add them up; 1,3,5 gives (1×1)+(3×2)+(5×3) = 22. Finally, multiply by 10 and add the number of letters in the word: (22×10) + 3 = 223. Now you go to the 223rd lightbulb and switch it on. After you've done this for every word in the dictionary, some of the lightbulbs are on and the rest are still off.

Now you are ready to do a spelling check. You take each word of the text and convert it to a number by the same process you used before; if you came across the word *ace,* you'd convert it to 223. You look at the appropriate lightbulb. If it's on, the word is acceptable; if it's off, it isn't. So, *ace* (lightbulb 223) is accepted. *Ade,* by contrast, would be converted to 243; the 243rd lightbulb would be off, so *ade* would be rejected.

The long line of lightbulbs is the 'bit map', an array of thousands of binary digits (0's and 1's). Converting a word to a number is known as hashing, and the method you use is a hashing function. The hashing function described above is too simple to do the job properly – *dcd, hdb* and various other non-words would all hash to 223 and be accepted – but it's possible to devise more complicated hashing functions so that hardly any non-words will be accepted. You may use more than one hashing function; you could derive, say, six numbers from the same word and check them all in the bit map (or in six separate bit maps), accepting the word only if all six bits were set.

If there is no need to save storage space, the dictionary can be held as a straightforward list of words, inflected forms included. The computer looks a word up in much the same way as a person looks one up in a printed dictionary. The words can be stored in alphabetical order, so the computer can go straight to the right place and check if it's there or not.

There are two ways in which a spelling checker can fail: it may flag a word as an error when in fact it's correct, or it may let an error slip through. Obscure words and proper names are the cause of the first problem. The frequency of these false alarms can be reduced by having a larger dictionary or a specialized one for particular kinds of text, such as a dictionary with lots of medical

terms, but there is no real solution to the problem of proper names – there are just too many of them. Many checkers have a facility whereby the user can build up a private supplement to the dictionary, to prevent the checker from constantly flagging names that crop up often in the user's documents.

False alarms, though irritating, may be acceptable in moderation since the user can always ignore the checker's output, but the second problem – letting errors slip through – is more worrying since the user cannot be sure that a passage is error-free even when the checker has gone over it. The problem arises because some misspellings match words in the dictionary, as in '*Their* she goes,' 'The *wether* was glorious,' or 'The Continental restaurant company is developing a chain of French-style *brassieres.*'[1] I call these 'real-word errors'.

Unfortunately the problem gets worse as the dictionary gets larger; including more obscure words in the dictionary, to reduce the number of false alarms, increases the risk of missing real-word errors. The word *wether* illustrates this. The word is, arguably, so obscure that any occurrence of *wether* in a passage is more likely to be a misspelling of *weather* or *whether* than a genuine occurrence of *wether,* so a checker that did not have the word in its dictionary would do better than one that did.

Drastic pruning of the dictionary, however, is not a solution; a checker with a small dictionary raises too many false alarms. A recent study has shown that, when an uncommon word occurs, it is far more likely to be a correct spelling of a rare word than a misspelling of some other word (Damerau and Mays 1989). This may not be true of some highly obscure words that resemble common words, such as *yor* and *stong,* so perhaps some judicious pruning is advisable. Nor is it true of certain medium-rare words that occur commonly as misspellings of other words, such as *cant* and *wont* which are often misspellings of *can't* and *won't;* these seem to require special treatment. But, with these provisos, big is beautiful for a checker's dictionary.

How serious is the problem of real-word errors? At first sight, it appears to be only marginal; the proportion of words that can be changed into another word by a small typing slip, such as *whether* into *wether,* is only about half of one per cent. However, the proportion is far higher among short, common words than among

long, rare ones. Mistyping *sat,* for instance, is quite likely to produce another word (*set, sit, sad* and so on), whereas mistyping *antirrhinum* is not. Taking this into account, the proportion of all typing errors that produce other words may be as high as sixteen per cent (Peterson 1986).

When spelling errors, as well as typing errors, are included, the problem becomes much more alarming. In the corpus of errors described in Chapter Four, forty per cent were real-word errors. In some cases the misspelling was based on pronunciation, and it was only by chance that it matched a dictionary word, such as *tort* for *taught,* but, more often, the misspelling was some other familiar word, as if the person writing it had two known spellings in mind and chose the wrong one. The wrong one was often a homophone of the right one, but not always. Errors of this kind were particularly likely to occur on function words (words like *of, and, be* and so on); in eighty per cent of the misspelt function words, the error consisted in writing some other function word in place of the one intended, such as '*He* name was Mrs Williams,' and 'You *we* treated like babies' (Mitton 1987).

Very few spelling checkers make any attempt to detect real-word errors, but at least three research projects have tried to tackle the problem. The first is a system called CRITIQUE (previously called EPISTLE) developed by IBM (Heidorn et al. 1982). This is a piece of software that will check the spelling, grammar and style of business correspondence. Armed with a complicated set of grammar rules for English, it attempts to parse each sentence of the text, i.e. to analyse a sentence into its syntactic parts – Noun (Subject of sentence), Adjective (qualifying Subject), Main Verb, Prepositional clause, and so on. If it fails, because the sentence is grammatically incorrect, it tries again, this time relaxing some of its grammar rules, and it carries on doing this until it achieves a successful parse. Since it knows which rule or rules it had to relax, it can work out what was grammatically wrong with the sentence. A real-word error quite often produces a grammatically incorrect sentence (such as 'I might of done'), so CRITIQUE can detect such errors and can sometimes suggest a correction, since the syntactic context gives a lot of clues to what the word should have been.[2]

The second project also depends on syntax as a way of spotting real-word errors. It is a modification of a system, developed at

Lancaster University, for tagging words in a text with their parts-of-speech (Marshall 1983, Garside et al. 1987). Given a sentence such as 'The fly bit the goat,' it first consults a dictionary to find out which tags (parts-of-speech) each of the words can have; it will find that *the* is a definite article, and that *fly* (likewise *bit*) can be a noun or a verb. It also has a table, derived from a large corpus of English text, showing the probability of a given tag being followed by another in a sentence; the table will show, for example, that a definite article is very likely to be followed by a noun, but not likely to be followed by a verb. It then works out, purely on the basis of probability, that *fly bit* in this sentence is likely to be Noun-Verb, rather than Verb-Noun (or Noun-Noun or Verb-Verb).

The system can be applied to looking for real-word errors by modifying it to report when it finds a sequence of tags that is very unlikely. For example, it would query 'Please complete the from in capitals,' since the sequence *the from in* (Definite article, Preposition, Preposition) has only a low probability (Atwell 1983, Garside et al. 1987).

Both these systems have some success in spotting real-word errors, but both tend to give too many false alarms because of sentences which are grammatically out of the ordinary but not ungrammatical (Richardson 1985, Leech et al. 1986). Neither, of course, can do anything about real-word errors that are not syntactically anomalous, such as 'We had thirty *minuets* for lunch,' 'We used to *pant* on Thursdays and hang up the *pitchers* on the walls,' 'There was a *fate* every summer.'

The third assault on real-word errors, again by researchers at IBM, resembles the Lancaster work somewhat in using probabilities derived from a very large corpus of text, but the probabilities are not of the co-occurrence of tags but of the co-occurrence of actual words (Mays et al. 1991). Given any two words from their 20,000-word dictionary, they can say what the probability is of any other of their dictionary words occurring next. Given, for instance, 'I think' as the first two words, they could say what the probability was of the word *that* occurring next. Or of the word *slowly* or *big* or *therefore* or *teapot*. (Presumably the probability of *that* after 'I think' is relatively high whereas the probability of *teapot* after 'I think' must be close to zero.)

In an experiment, they took sentences containing a single real-word error, such as 'The thief licked the lock,' (for *picked*). The misspellings were all of the simple typing-slip kind, i.e. differing by just one mistype from the correct spelling. (I explain below what I mean by that.) When considering a word as a possible error, the system first generated all the words that might have been changed into this word through a typing slip. For example, from *licked* it would have generated *kicked, ticked, locked, liked* and so on, including *picked.* For each of these alternatives, it calculated the probability of the whole sentence from its table of three-word probabilities, i.e. one value for 'The thief *kicked* the lock,' another for 'The thief *ticked* the lock,' and so on. It also calculated the probability of the original sentence, 'The thief *licked* the lock.' If 'The thief *picked* the lock' came out as more probable than 'The thief *licked* the lock,' it would conclude that *licked* was a real-word error that should be corrected to *picked.*

It could be wrong either by leaving the original error uncorrected or by preferring the wrong alternative or by 'correcting' some other word in the sentence. It had no way of knowing in advance that *licked* was the misspelling here. It would go through the same procedure with all the other words. It would generate *dock, rock, sock, look* and so on for *lock* and might possibly prefer 'The thief licked the *rock.*'

There was a further factor in its calculations, namely a general level of expectation of errors in the text. This was set by the experimenters at levels between 0.1 and 0.0001. Essentially, if it was told to expect a lot of errors, it tended to make a lot of corrections, i.e. to rate the alternatives as more probable than the original, though many of its 'corrections' were wrong. If it was told that errors were rare, it was more respectful of the original text; when it did make a correction, it was nearly always right, but it left a lot of the misspellings uncorrected.

It is not clear what application this method could have to ordinary spelling checkers in the near future because of its considerable demands on memory and computing power, but it is the only method I know of that has been capable of detecting (and correcting) syntactically acceptable real-word errors in unrestricted text.

Spelling correction

Many people find that a spelling checker is all they need; they know how to spell and they just want their occasional slips to be pointed out to them. People who have trouble with spelling, however, need something more. Suppose you have written *neumonia* and the checker has told you this is wrong. If you don't know how to spell *pneumonia,* you're stuck. The dictionary is no help. You want the computer to tell you the correct spelling.

To correct someone's spelling errors, you have to be able to guess what words the person meant and you have to be able to spell them correctly. People generally find the first part easy but the second part harder; most people would understand 'She was excused swimming because of her verouka,' but they would not be able to correct it. For computers, it's the other way round. Producing a correct spelling is easy – they can store a complete dictionary and retrieve any word as required; the hard part is deciding which word was intended.

It is for this reason, incidentally, that one cannot say in general whether computers are better or worse than people at spelling correction. Given a minor misspelling of a long word, such as *innoculation,* a computer will detect it and correct it better than most people would, because it is easy to guess what word was intended but not easy to spell it. By contrast, with a misspelling of a common word, such as *cort* ('We got cort in the rain'), a computer might have difficulty deciding that *caught* was the word intended, whereas most people would correct it easily.

Given a dictionary of realistic size – say 30,000 to 80,000 words – it is not practical to go through the entire dictionary for each misspelling, considering every word as a possible candidate; a corrector has to select a section of the dictionary, of some tens or hundreds of words, and search through these in the hope of finding the correct word.

Analyses of errors – mainly typing errors – in very large text files (Damerau 1964, Pollock and Zamora 1984) have found that the great majority of wrong spellings (eighty per cent to ninety-five per cent) differ from the correct spellings in just one of the following four ways:

one letter wrong (*peaple*)
one letter omitted (*peple*)
one letter inserted (*peopple*)
two adjacent letters transposed (*pepole*)

It has also been found (Yannakoudakis and Fawthrop 1983a) that the first letter is usually correct. Given a mistyped word, therefore, there is a good chance that the correct spelling will begin with the same letter and will be either the same length or just one letter longer or shorter. If the words are held in order of first letter and length, it is easy for the corrector to restrict its search to the appropriate section of the dictionary (Turba 1982).

Words that are misspelt, as opposed to mistyped, tend to differ from the correct spellings in more than just the simple ways listed above (Mitton 1987). For example, *disapont* – a misspelling of *disappoint* – is two letters shorter than the correct word; looking through the dictionary at words beginning with *d* and of seven to nine letters long would fail to find *disappoint.* You could simply increase the number of words to be considered, perhaps taking in words that are two letters longer or shorter than the misspelling, but this would increase substantially the number of words the corrector had to look at, so it would take longer to produce its correction. It would also be inefficient since a large proportion of the words it looked at would be nothing like the misspelling; for *disapont,* it would take in *donkey* and *diabolical,* which are obviously not what *disapont* was meant to be. What is needed is some way of retrieving those words that have some resemblance to the misspelling.

This problem has been around for a long time in the context of retrieving names from a list of names. Suppose you are working at an enquiry desk of a large organization, with a terminal connecting your office to the central computer. A customer comes in with a query about her account. She says her name is *Zbygniewski.* You don't want to ask her to spell it – perhaps her English is poor and other customers are waiting. To make matters worse, the name may be misspelt in the computer file. You want to be able to key in something that sounds like what she just said and have the system find a name that resembles it.

The Soundex system was devised to help with this problem (Knuth 1973, Davidson 1962). It dates, in fact, from the days of card-indexes – the name stands for 'Indexing on sound' – but has been transferred to computer systems. A Soundex code is created for every name in the file. I will present the details in Chapter Nine, but the idea of the code is to preserve, in a rough-and-ready way, the salient features of the pronunciation. Vowel letters are discarded and consonant letters are grouped if they are likely to be substituted for each other – an *s* may be written for a *c*, for instance, but an *x* for an *m* is unlikely.

So, every name in the file has one of these codes associated with it. The name *Zbygniewski* has code Z125, meaning that it starts with a *Z*, then has a consonant in group 1 (the *b*), then one in group 2 (the *g*) and then one in group 5 (the *n*), the remainder being ignored. Let's say you key in *Zbignyefsky.* The computer works out the Soundex code for this and retrieves the account details of a customer with the same code – *Zbygniewski* – or perhaps the accounts of several customers with somewhat similar names.

It is fairly obvious how this system can be applied to spelling correction. Every word in the dictionary is given a Soundex code. A Soundex code is computed from the misspelling, and those words that have the same code are retrieved from the dictionary. *Disapont* would produce the code D215, and the set of words with code D215 would include *disappoint.*

A similar system was devised by the SPEEDCOP project mentioned in the last chapter (Pollock and Zamora 1984). A key was computed for each word in the dictionary. This consisted of the first letter, followed by the consonant letters of the word, in the order of their occurrence in the word, followed by the vowel letters, also in the order of their occurrence, with each letter recorded only once; for example, the word *xenon* would produce the key *XNEO* and *inoculation* would produce *INCLTOUA.* The words in the dictionary were held in key order, as illustrated by the small section shown in Figure 7.1. (The purpose of the SPEEDCOP system was to correct a database of scientific text, hence the inclusion of many technical terms in the dictionary.)

PLTDOE	plotted
PLTE	pellet
PLTEI	pelite
PLTIO	pilot
PLTNGAI	plating
PLTNSUO	plutons
PLTNUO	pluton
PLTOU	poult

Figure 7.1 A section of the SPEEDCOP dictionary

When the system was given a misspelling, such as *platin,* it computed the key of the misspelling and found its place in the dictionary. In this example, the key of *platin* would be *PLTNAI,* which would come between *PLTIO* and *PLTNGAI.* Moving alternately forwards and backwards from that point, it compared the misspelling with each of the words to see if the misspelling could be a single-error variation on that word, until either it had found a possible correction or had moved more than fifty words away from its starting point. The SPEEDCOP researchers found that, if the required word was in the dictionary, it was generally within a few words of the starting point. In the example, the corrector would quickly find the word *plating* as a possible correction (*platin* being an omission-error variant of *plating*).

The Soundex code and the SPEEDCOP key are ways of reducing to a manageable size the portion of the dictionary that has to be considered. Confining the search to words of the same length (plus or minus one) restricts the search even further. The price to be paid is that, if the required word is outside the set of those considered, the corrector is not going to find it.[3]

The next task facing the corrector is to make a selection from the words it looks at – a best guess, or at least a shortlist. If the corrector is intended mainly to handle typing errors, this task is not difficult. Given that the great majority of mistyped words fall into one of the four classes listed above, the corrector compares the misspelling with each candidate word from the dictionary to see if they differ in one of these four ways. If they do, then that candidate joins the shortlist. Given the misspelling *brun,* for

instance, the corrector would produce the list *brunt* (omitting one letter gives *brun*), *bran* (changing one letter), *bun* (inserting one letter) and *burn* (transposing adjacent letters).

Another way of selecting candidates is to calculate, in some way, how closely each word resembles the misspelling and to shortlist those that have the best scores. This process is called 'string-matching', and there are many ways of doing it. One way is to see how many chunks of the shorter string are present in the longer string (Joseph and Wong 1979). For instance, given *medsin* and *medicine,* you could say that *medsin* has the *med* and the *in* of *medicine,* a total of five letters out of the eight in *medicine,* a score of sixty-three per cent. Another method considers the number of trigrams (three-letter sequences) that the two strings have in common (Angell et al. 1983). *Medicine* and *medsin* would be divided up as follows (the # symbol marks the beginning or end of a word):

medicine	#me med edi dic ici cin ine ne#
medsin	#me med eds dsi sin in#

The more trigrams the two have in common, the better match they are considered to be. Some methods give more weight to letters near the front; others rate letters near the end more highly than those in the middle; some rate certain letters more highly than others, such as consonants over vowels.[4] Some hand-held spellcheckers make use of a special-purpose chip which implements string comparisons at high speed (Yianilos 1983).

A project at Bellcore is investigating the use of spelling correction in an unusual setting, namely to assist deaf or speech-impaired people to use the telephone (Kukich 1992b). Deaf people can communicate with each other over a telephone line by using a screen and keyboard. When they want to converse with a user of a voice telephone, they go via a relay centre. The voice user speaks to the relay person who types the message to the deaf person; the deaf person types back and the relay person speaks it. Bellcore would like to automate this process and part of this involves the generation of computer speech from the keyed text. But this text typically contains typing errors which upset the speech generator, hence the need for spelling correction. The corrector is allowed to make only one correction for each misspelling, not a list of possible

corrections such as a spellchecker would produce.

Experiments have found that one of the simpler methods is the most effective. A 'feature vector' of about five hundred bits (think of a line of lightbulbs again) is computed for each word in the dictionary. If the word contains an *a*, the first bit is set (the first lightbulb is turned on); if it contains a *b*, the second is set, and so on. If it contains *aa*, the 27th is set; if it contains *ab*, the 28th is set. (There is no place in the line for letter-pairs that don't occur in English, such as *yy*.) A corresponding feature vector is computed for the misspelling and this is compared with the vectors of the dictionary words. The word whose vector is most like the misspelling's vector (most nearly has its lightbulbs on and off in the same places) is chosen as the correction.

Some methods of string-matching make use of tables showing the likelihood of this or that letter being involved in an error. I describe one of these methods in more detail in the next chapter (Wagner and Fischer 1974). It was developed for correcting the output of an optical character reader. These machines are prone to make certain errors more than others; for example, they are likely to read an *e* as an *o*, but not likely to read a *t* as an *m*. The corrector has a table showing the probability of one letter being mistaken for another, and it uses these figures in deciding what the word ought to be. Given *gom*, it would guess that the word was *gem* rather than *got*.

Probability is also the basis of an approach developed at Bell Labs for correcting typing errors (Kernighan et al. 1990, Church and Gale 1991). This system has tables of error probabilities derived from a corpus of millions of words of typewritten text. The tables give the probability of an *a* being substituted for a *b*, a *p* being inserted after an *m*, and so on. It also has an estimate of the probability of any particular word occurring in the text.

When it detects a misspelling (which it does by dictionary look-up), it first retrieves from the dictionary all the words that could have given rise to this misspelling by a single mistype. (It doesn't handle more complicated errors.) For example, from the misspelling *acress*, it retrieves *actress*, *cress*, *caress*, *access*, *across* and *acres*. Taking *actress*, it consults its table for the probability of having a *t* omitted after a *c* and combines this with the probability of meeting the word *actress*. In this way it produces a probability estimate for

each of the candidates and it then puts the candidates in order of probability for presentation to the user.

The errors that poor spellers make are more complicated than those of an optical character reader or a typist, but a similar approach can still be used. One system (Yannakoudakis and Fawthrop 1983b) has a table of error-patterns, derived from the analysis of a corpus of spelling errors; the table might show, for instance, that *au* is sometimes written as *or,* or *ch* as *tch.* It compares the misspelling with each of the words in the section of the dictionary that it's looking at to see if the difference follows the patterns in its table. For example, given *lorntch,* it would find that *launch* differs from it in two of these ways. The table also contains information about the frequency with which each of these error-patterns occurs, so the corrector can put the shortlisted candidates into order. When trying to correct *lorntch,* it would also find *lounge* but it would rate this as less likely than *launch* because the table contains the information that *or* for *ou* and *ge* for *ch* are less likely than *or* for *au* and *tch* for *ch.*

Some of the more advanced commercial correctors also retrieve candidates on a 'phonetic' basis. Their dictionaries presumably contain information about pronunciation, and the correctors use this to offer words that might sound like the misspelling, even though they don't look much like it; for *newmoanya,* for example, their list would include *pneumonia.*

Commercial companies tend not to publish details of how their spellcheckers work, but there is one pronunciation-based spell-checker described in the research literature; it was developed in the Netherlands for the correction of Dutch, though the principles would apply to English also (Van Berkel and De Smedt 1988). It uses a variation on the trigram system mentioned earlier, but with pronunciations rather than spellings. Given the misspelling *indissceat,* for example, it would begin by making a guess at the pronunciation – perhaps /ɪndɪski:t/ – then break this up into 'triphones' and then compare this with the pronunciations of various words in its dictionary, also broken up into triphones. The comparison with *indiscreet* would look like this:

indissceat	#ɪn ɪnd ndɪ dɪs ɪsk ski: ki:t i:t#
indiscreet	#ɪn ɪnd ndɪ dɪs ɪsk skr kri: ri:t i:t#

The more triphones a dictionary word has in common with the misspelling, the better match it is considered to be.[5] Homophones, of course, match perfectly.

Most correctors simply offer a small selection of possible corrections, generally about six, for the user to choose from, though some correctors offer dozens of suggestions if the user wants them. This shortlist, however, is often a curious rag-bag of words. When asked to make suggestions for *perpose,* Microsoft Word Version 6.0 produced the list (in this order) *preppies, propose, papoose, prepuce, preps* and *props,* but not *purpose.* The lists often contain obscure words with no indication of their level of obscurity; many of the offerings are wildly inappropriate for the context and perhaps not even syntactically possible. When asked for suggestions for *cort* in 'I've cort a cold,' Wordperfect 5.1 produced – take a deep breath – *cart, cert, coat, colt, cont, coot, copt, cor, cord, core, corf, cork, corm, corn, corp, corr, cors, corti, cost, cot, court, crt, curt, carat, carate, card, cared, caret, carried, carrot, carte, cerate, cered, ceroid, chaired, charade, chard, chariot, charred, chart, cheered, cheroot, chert, chirred, chord, choreoid, chorioid, choroid, cirrate, cored, corrade, corrode, corrupt, coward, cowered, curate, curd, cured, curet, curette, curried, karate, kart, keyword, scared, scarred, scirrhoid, sciuroid, scored, scoured* and *scurried,* but not, alas, *caught* (perhaps because *caught* and *cort* are not homophones in American speech). One can't help feeling that the corrector ought to be able to do better – to restrict its list to plausible suggestions and to order them so that its best guess is generally the one required. Given 'You shud know,' it ought to offer *should* ahead of *shad* and *shed.*

Word frequency can help; *shad* could be removed from the above list, or at least relegated to the bottom, purely because of its rarity. But it doesn't help much; candidates in the shortlist are often of similar frequency, such as *there* and *their* for *ther,* and a rare word will occasionally be the one required.

Syntax can also help. I described earlier how some correctors do a syntactic analysis in order to spot real-word errors; they can use the same analysis to rule out some of the candidates. Quite often, as in *shad, shed, should,* there will be only one candidate left.

A semantic analysis is much more difficult for a computer to attempt, but it may be possible when the subject matter of the text

is restricted (Morgan 1970, Teitelman 1972). For example, a corrector that checked the commands that people typed into an electronic mail system would be able to correct *Snd* to *Send* (rather than *Sand* or *Sound*) in 'Snd message to Jim,' because *Send* is one of the few words that could occur at that point in this sentence (Durham et al. 1983). Similarly, a system that handled enquiries to British Rail would be able to use its interpretation of the meaning to correct 'Is there an erlier conexson?' (Hendrix et al. 1978) A system of this kind might be able to detect some real-word errors. A computerized tourist guide might detect that a query about *gold courses* was really about *golf courses.* More ambitiously, a system that conducted a dialogue with a user might be able to build up a representation of what the user had in mind and use this for spellchecking (Ramshaw 1994). If a user of the computerized tourist guide had been asking about holidaying in the west country and then asked 'Are there trains to *Swinton*?' the system might guess that he meant *Swindon,* since Swindon is on the main line from London to the west whereas the places called *Swinton* are all in the north. In general, however, spellcheckers that handle unrestricted text do not have enough information about the words in their dictionaries or about the topics people write about to enable them to make any use of the semantic context.

At present, then, checkers and correctors play a small but useful role in helping people to remove minor errors from their written work. Some systems are just checkers – they flag errors but make no attempt to offer suggestions – and this is often all that is required; if you've typed *adn* for *and,* you can correct it easily. Most systems, however, do both checking and correcting, so that the word *spellchecker* usually means a piece of software that both checks the text and offers suggestions for misspelt words. A list of suggestions can occasionally be helpful, especially for people whose spelling is a little weak; not everyone would know, if a checker queried *occurence,* that it ought to be *occurrence.* But spellcheckers are still some way short of offering the help that a poor speller wants – the kind of job that a good typist would do.

They miss a fairly high proportion of errors; real-word errors form a substantial minority of spelling errors and most spellcheckers ignore them completely. Their suggestions are often irritatingly inappropriate, frequently including words that are

obscure or syntactically out of place. If the misspelling differs from the correct word in certain ways, such as having a different first letter (*nowledge, wrankle, eny*), or being more than one letter longer or shorter (*probly, cort, unforchunitley*), or having several letters different (*payshents, powertree, highdrawlick*), the required word may not be in the list of suggestions at all.

Notes

1. The *brassieres* (for *brasseries*) error was in a report quoted in a short piece about spellcheckers in The Times of 16 February 1995.

2. A similar system has been implemented in a language-sensitive text editor for Dutch (Kempen and Vosse 1992). It is capable, for example, of detecting the misspelling *word* in *Peter word bedankt* (English *Peter am thanked*) and correcting it to *Peter wordt bedankt* (*Peter is thanked*).

3. The SPEEDCOP researchers found that the most frequent cause of failure with their system was the omission of consonant letters near the beginning of a word (Pollock and Zamora 1984). For example, the misspelling *pating* would produce the key PTNGAI, which might be some distance away from PLTNGAI, the key of *plating*. They therefore computed a second key, called the 'omission key'. They knew from their analysis of a large corpus of spelling errors that consonant letters were omitted in the following order of increasing frequency – the letter *j* was omitted the least and the letter *r* the most:

   J K Q X Z V W Y B F M G P D H C L N T S R

   The omission key consisted of the consonant letters of the word sorted in this order, followed by the vowel letters in their order of occurrence in the word. The omission key for *pating* would be GPNTAI, which would probably be close to the omission key for *plating* – GPLNTAI.

4. Some of these variations are described in Hall and Dowling (1980), Alberga (1967), Blair (1960) and Cornew (1968).

5. For some misspellings it is possible that more than one variant pronunciation might be generated, though many details of the pronunciation, such as stress pattern, can be ignored since this application is less demanding than, say, speech generation. The dictionary also stores the frequency with which each triphone occurs and these frequency values are taken into account; if two pronunciations share an unusual triphone in common, this will be considered more significant than if they share a run-of-the-mill one.

CHAPTER EIGHT

Generating a list of suggestions

The previous chapters have described some of the problems that spellcheckers have to face, the methods that they use for tackling these problems and the extent to which they fall short of doing the job that people would like them to do. The next three chapters describe a spellchecker which attempts to address some of these problems and to improve on the performance of current commercial spellcheckers, especially when faced with text written by poor spellers. This spellchecker is the product of my own research and it resides currently on the Birkbeck College computer. It does not have a name so I refer to it simply as 'the prototype' or 'the corrector'. First, a sketch of how it looks to the user.

The corrector takes a piece of text and goes through it sentence by sentence, calling the user's attention to any words that it thinks are misspelt and, on request, offering a short list of possible corrections, ranked in order of best-guess, second-best and so on. For example, given the following sentence:

On Wenzdays, their was speling and mutliplucation.

the corrector would query the words *Wenzdays, their_was, speling* and *mutliplucation.*

The corrector has two levels of query. A serious query indicates that this word is not in the corrector's dictionary and is therefore likely to be a misspelling; a more tentative query means that the corrector has some other reason for thinking this may be an error, or is anyway not so sure about it. In the above sentence, the words *speling* and *mutliplucation* are given the first-level query; it is likely that these are misspellings. The other two receive the more

tentative query. The word *Wenzdays* is not in the dictionary, but it begins with a capital letter so it could be a name. *Their* and *was* are both dictionary words, but the combination is strange; the underscore in *their_was* indicates that it is the combination which is suspect.

For each word queried, the user may keep the word as it is, accept one of the corrector's suggestions, or type in some other word. The lists of suggestions for *Wenzdays, their, speling* and *mutliplucation,* would be as follows:

> Wenzdays: 1 Wednesdays 2 Wednesday 3 Weekdays
> their: 1 there 2 other 3 tear 4 theory
> speling: 1 spelling 2 spilling 3 spoiling 4 spieling
> mutliplucation: 1 multiplication 2 multiplications

If the misspelling begins with a capital, like *Wenzdays,* the suggestions also begin with capitals. If the misspelling ends with an apostrophe or *'s,* the suggestions end in the same way unless they are the kind of word (such as a verb) that cannot take an apostrophe.

If the word has an initial capital, and it's not the first word in the sentence, and the user opts to keep it, then the corrector takes it to be a name and doesn't query it if it crops up again in this passage. Most commercial spellcheckers allow the user not only to keep a word unaltered in the passage being corrected but also to include it in a private dictionary so that the spellchecker will refrain from querying it in the future. This facility is missing from the prototype, but including it would not present any serious problems.

So much for what the user sees. The next three chapters describe what goes on, as it were, under the bonnet. Figure 8.1 presents a simplified overview of the program.

I will describe the corrector from the inside out. This chapter and the next are devoted to the part described in the program as 'Generate list of suggestions'. In this chapter I will assume that the corrector has already decided somehow that a particular word in the input text is a misspelling; this is most likely to be a non-word, but it could be a real-word error like the *their* of *their_was.* I am also assuming that it has retrieved a set of, say, a hundred words from its dictionary which look reasonably promising. The task

```
Taking the text sentence by sentence:
    Split the input into words and store each word in memory.
    Look up each word in the dictionary, and mark it if not found.
    Check each pair of words for anomalies of syntax.
    Display sentence, possibly with queries.
    If any words have been queried, then
        For each queried word, do:
            Generate list of suggestions.
            Offer best few to user.
            Get user's decision.
            Insert user's choice in sentence.
```

Figure 8.1 A simplified overview of the program

now is to whittle these down to a shortlist of about five and to rank these in order. The next chapter will describe how the set of reasonably promising words gets selected from the dictionary, and the one after that will describe how the corrector decides which words to query.

For example, from the sentence, 'Do you like spaggety?' the corrector has decided that *spaggety* is a misspelling and has retrieved *spigot, spiked, sparked, spectre, sprigged, specked, sprocket, sobriquet, sprightly, spaghetti* and many more. Now it has to choose the best five of these and, hopefully, to put *spaghetti* at the top.

It does this by matching the misspelling against each of the candidates in turn and computing a score for each one. This score represents, in a way, the distance between the misspelling and the candidate. *Spaghetti,* one might say, is quite close to *spaggety* whereas *spectre* is rather more remote, so *spaghetti* ought to get a low score (indicating closeness) while *spectre* should get a high one. The candidate with the lowest score is the best guess.

Traversing a directed network

The score for each candidate is computed by a dynamic programming algorithm that treats the string-matching as the problem of traversing a directed network.[1] I will first describe the

general algorithm and then I will show how it can be used for string-matching. Figure 8.2 shows a directed network.

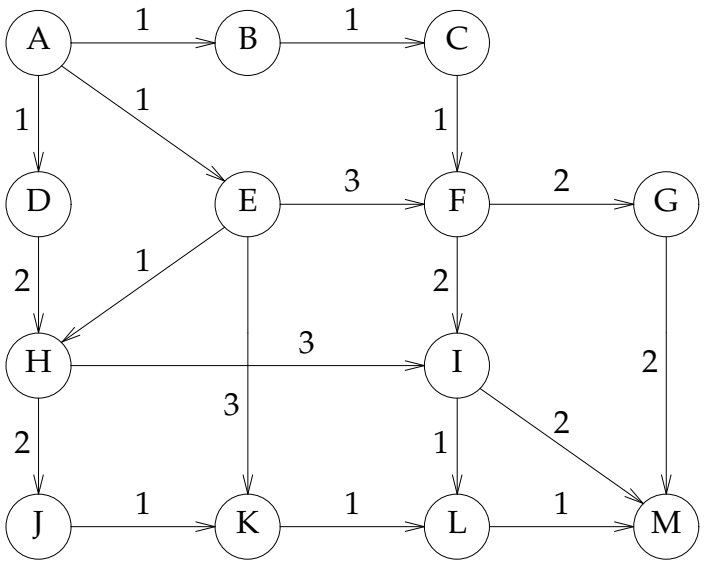


Figure 8.2

A directed network is simply a set of nodes (the circles) joined by arcs (the arrows). For the algorithm to work, the network has to be acyclic and the nodes have to be topologically ordered. To say that it is acyclic means that, if you set out from any node and travel along the arcs (in the direction indicated by the arrows), it is impossible to return to the node you started from. The network of Figure 8.2 is acyclic. To say that the nodes are topologically ordered means that the nodes can be ordered in some way and that, if an arc goes from node 1 to node 2, node 2 is always further on in the order. In Figure 8.2 the nodes are labelled with letters so they can be put into alphabetical order and the arcs all point to nodes further on in the alphabet, so they are topologically ordered. (If an arc were added from K to I, the network would no longer be topologically ordered; if arcs were added both from K to I and from I to E, it would no longer be acyclic.)

The numbers on the arcs are costs. Let's suppose the costs are in pounds, so that it costs one pound to get from A to B, two pounds

to get from D to H, and so on. The problem is to find the lowest cost of getting from the start-node A to the end-node M. Route A-B-C-F-G-M, for instance, costs seven pounds; route A-E-F-I-M costs eight. What is the lowest possible cost?

The algorithm takes the nodes in alphabetical order and computes, for each node, the lowest cost of getting to that particular node. Starting with a cost of zero for A, it obviously costs 1 to get to B, 2 to get to C, 1 to D and 1 to E. With node F we can see the algorithm beginning to work because now we have a choice. We can get to F either from C, at a cost of 2+1=3, or from E at a cost of 1+3=4, so we choose the cheaper; we can get to F for 3. That means it costs 5 to get to G (3+2). For both H and I we have to make a choice, and so on. It turns out that the cheapest way of getting from A to M will cost 6 (the route is A-E-K-L-M).

Now to return to string-matching. Imagine that you are speaking to a friend on a very bad telephone line and that he is spelling out a name for you letter by letter. For most of the letters, you hear something or other, but not always clearly enough for you to guess correctly what the letter was; sometimes you miss a letter completely and sometimes you mistake one of the extraneous noises on the line for a letter and insert it into the name. The name might be *Birkbeck* but you write *Pirbeack.*

Your friend is providing an input string and you are producing an output string letter by letter, and, at each stage, one of three things can happen. He can move forwards one letter in the input string without you producing any output letter (omission), or you can produce an output letter without him moving forwards through the input string (insertion), or he can move forwards one letter in the input string and you produce an output letter (substitution). Getting a letter right is a special case of substitution; if the input letter is *k* and the output letter is *k,* you are substituting a *k* for a *k.*

This description would also apply to an optical character reader as it scans a word from left to right, making a guess at each letter. At each stage it might skip over an input letter without producing an output letter (omission), or produce an output letter without taking any of the input (insertion), or it might produce an output letter for an input letter (substitution). (Actually, these devices make very few insertion or omission errors,[2] but I'll keep all three

types of error in the example for completeness.) Character readers often incorporate a spelling checker and a corrector to monitor the output of the scanner and, if possible, to correct its mistakes.

Suppose one of these character readers has scanned a word and the scanner has produced *plog* as its output. The checker ascertains, by dictionary look-up, that *plog* is not correct, and hands it on to the corrector. The corrector retrieves a number of words that look something like *plog – peg, plug, plague* and so on – and is now trying to decide which of these is most likely to be the word that has just been scanned.

It takes the words one by one; let's suppose it is considering *peg*. Given that the scanner might have done any of the above three things (substitute an output letter for an input letter, omit an input letter, insert a letter into the output) at each stage as it moved through *peg*, there are many ways in which it could have read *peg* but produced *plog*. These can be represented as paths through a directed network, as shown in Figure 8.3.

The horizontal arcs correspond to insertions, the vertical arcs to omissions and the diagonal ones to substitutions. Any path from A to T corresponds to one way in which the scanner might have scanned *peg* and produced *plog*. For example, the route A-I-J-O-T corresponds to scanning a *p* and outputting a *p* (A-I), then inserting an *l* (I-J), substituting an *o* for an *e* (J-O), and finally scanning and producing a *g* (O-T). (Note that diagonal arcs correspond to both correct and incorrect substitutions.)

There are, however, many other paths, and these correspond to other ways, albeit unlikely ones, in which the scanner might have scanned *peg* and produced *plog*. For example, the route A-B-C-D-Q-R-S-T corresponds to the remote possibility that it might have omitted all three letters of *peg* one after another (A-B-C-D) and then produced, without taking any more input, the letters *plog* (D-Q-R-S-T), the resemblance between *plog* and *peg* being purely coincidental.

It is obvious to us, of course, that most of these other paths are very unlikely, but it is not obvious to the corrector unless it is provided with some more information. What it needs to know is the probability of the scanner taking each of these arcs (i.e. performing each of these operations). If it knew that the scanner usually read letters correctly and only rarely inserted or omitted

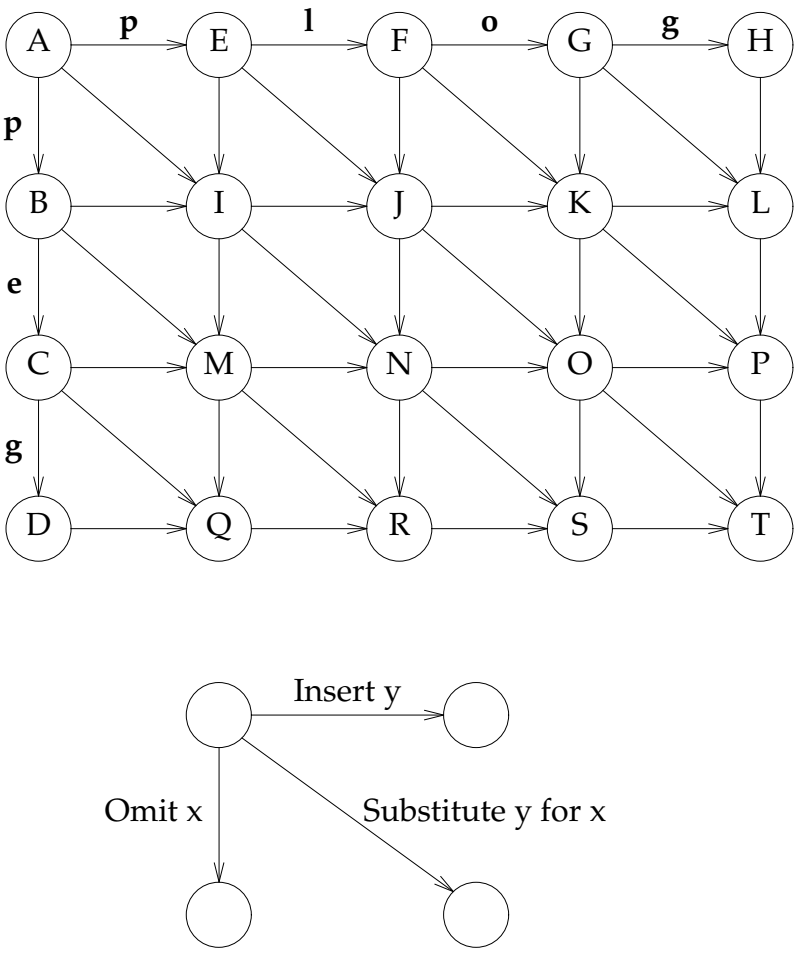


Figure 8.3

letters, it would know that, say, A-I was more likely than A-B or A-E. Figure 8.4 presents the *peg*-to-*plog* network again with this information added, not actually in the form of probabilities but in the form of costs. Each arc has a cost between nought and five where nought corresponds to 'probable' and five to 'unlikely'. For example, substituting an *o* for an *e* (arc J-0) is considered quite likely, so it is given a low cost – just 1, whereas substituting a *g* for an *e* (arc K-P) is considered most unlikely and is given a high cost – 5. (The next section considers how these costs are determined.)

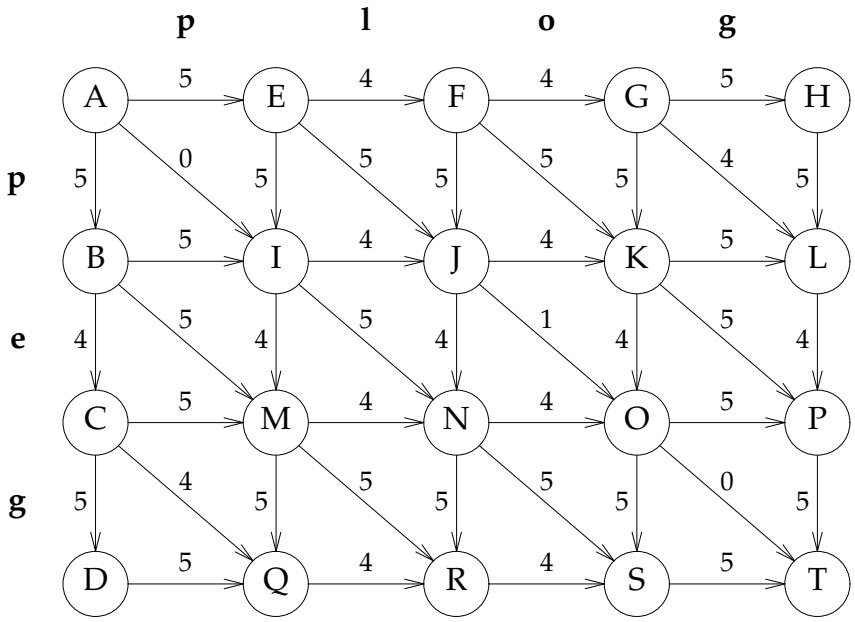


Figure 8.4

This network is both acyclic and topologically ordered, so the algorithm described earlier can be used to compute the minimum cost of traversing it; the answer is five. The same procedure would be applied to *plug, plague* and the rest, producing a score for each one – perhaps two for *plug,* ten for *plague* and so on. The lower the score, the more promising the candidate; the corrector is saying that *plug* is a closer match for *plog* than either *peg* or *plague* – that *plug* is more likely than *peg* or *plague* to have been misread as *plog.* It does this for all the candidates and chooses the one with the lowest cost as its best guess.

The programming of this string-matching algorithm is fairly straightforward. (Readers unfamiliar with computer programming may skip to page 119.) I present it in a Pascal-style language, and, for clarity, I assume that both the misspelling and the candidate-word are in lower-case letters with no hyphens, apostrophes or any other such characters. The arc-traversal costs are held in three arrays called Om (omission costs), Ins (insertion costs) and Sub (substitution costs).

```
Om, Ins : array ['a'..'z'] of 0..5;
Sub : array ['a'..'z', 'a'..'z'] of 0..5;
```

For example, if the cost of omitting a *p* is to be set to 5, as in Figure 8.4, this will be represented by setting the value of Om['p'] to 5. Ins['l'] will hold the cost of inserting an *l* i.e. 4. Sub['o','e'] will hold the cost of putting an *o* in place of an *e* i.e. 1 (corresponding to arc J-O in the diagram). Sub['p','p'] (arc A-I) and Sub['g','g'] (arc O-T) will hold the value zero, as you would expect.

The two strings are Word and Misp (candidate-word and misspelling) and their lengths Wordlen and Misplen. The nodes of the network are represented as an array called Cost, in which the value at an element will represent the minimum cost of reaching that node.

```
function mincost : integer;
var   Cost : array [0..Wordlen, 0..Misplen] of integer;
      i : 1..Wordlen;  j : 1..Misplen;
      arcsub, arcom, arcins : integer;
begin
    Cost[0,0] := 0;
    {Assign values to nodes reached by only one arc.}
    for i := 1 to Wordlen do
        Cost[i,0] := Cost[i–1,0] + Om[Word[i]];
    for j := 1 to Misplen do
        Cost[0,j] := Cost[0,j–1] + Ins[Misp[j]];
    {Assign values to nodes reached by three arcs.}
    for i := 1 to Wordlen do
        for j := 1 to Misplen do
        begin
            arcsub := Cost[i–1,j–1] + Sub[Word[i],Misp[j]];
            arcom := Cost[i–1,j] + Om[Word[i]];
            arcins := Cost[i,j–1] + Ins[Misp[j]];
            Cost[i,j] := min(arcsub,arcom,arcins)
        end;
    mincost := Cost[Wordlen,Misplen]
end;
```

Taking the *peg* to *plog* network of Figure 8.4 as an example, the function first assigns zero to node A, then assigns values to the

nodes down the left-hand side (B-C-D), then those across the top (E-F-G-H), and then does the rest in alphabetical order. Taking node J as an example, arcsub is the value at E (five) plus the cost of the E-J arc, i.e. Sub['p','l'], which is five, making ten; arcom is the value at F (nine) plus Om['p'] (five), making fourteen, while arcins is the value at I (zero) plus Ins['l'] (four) making four. The lowest of these three values is the four, so four is assigned to node J. Eventually a value is assigned to Cost[Wordlen,Misplen] (node T in the example), and this is the minimum cost of traversing the network.[3]

Putting costs on the arcs

Returning now to the problem of correcting misspellings made by people, the same technique could be used to produce scores for the various candidates for *spaggety;* directed networks could be set up and minimum traversal costs computed. But would the best shortlist be selected and would the best candidates be ranked in the right order?

This depends on the scores, and the scores depend entirely on the costs assigned to the arcs, and this brings us up against an important difference between optical character readers and human beings. It would be possible to run a scanner over a large amount of text and to analyse its output, tabulating the number of times it inserted a *z* or omitted an *l* or substituted an *e* for an *o* and so on. The resulting tables would form the basis for the costs on the arcs. If, as is likely, the scanner often output an *e* for an *o,* then a low value would be assigned to Sub['o','e']; conversely, if it never output an *x* for a *p,* then a high value would be placed in Sub['p','x'].

By contrast, it would be fairly pointless to take a corpus of text written by people and to tabulate all the single-letter mistakes in it. The letter *k,* it might turn out, is sometimes omitted. At first sight, then, it might seem that a lower-than-average cost should be assigned to Om['k'], for the correcting of people's misspellings. But closer inspection would show that people do not have, as it were, a stochastic propensity for omitting a *k* now and again; it's

rather that they are likely to omit certain *k*'s in certain places. They might omit the leading *k* of *knickers* and *knuckles* but not of *kettle* and *kitchen.* Similarly, a single-letter analysis might show that an *s* is sometimes written for a *c,* but it does not follow that a low value should be assigned to Sub['c','s']. The *c* of *cement* may sometimes be written as an *s,* but not the *c* of *cat.*

A scanner operates on a given letter in much the same way regardless of the letters on either side, but the things that people are likely to do with a letter (omit it, insert it, confuse it with some other letter) depend very much on the words they are trying to spell. When trying to correct the output of a scanner, the corrector needs only one set of Om, Ins and Sub values for all words. When trying to correct misspellings made by people, a corrector needs a different set of values for each candidate word. (I am speaking here about spelling errors rather than typing slips; a table of error frequencies built from a large corpus of typing errors might well be useful in correcting typing slips.)

I pointed out in Chapter Five how the misspellings of a given word have a sort of family resemblance, the most common misspellings resembling the correct spelling quite closely with the rarer ones having only a distant resemblance. We would regard *sissors* as a close relative of *scissors* but would feel that *satter* was rather more remote from *scatter;* we make allowances for the missing *c* of *scissors* but are not prepared to make such allowances for the *c* of *scatter.* For the corrector to make the same sort of judgement, i.e. to produce a low score (indicating closeness) for *sissors/scissors* but a higher one for *satter/scatter,* it has to use a slightly different Om table for *scissors* (one with a low value for Om['c']) than it would for *scatter.*

The solution adopted in the prototype corrector is to have a basic set of values in the Om, Ins and Sub tables, but to store information in the dictionary about how these values should be changed when considering a word as a candidate. For example, the entry for *scissors* contains the information that a low value should be used for Om['c']; the entry for *pouch* says that a low value should be used for Ins['t'] (to anticipate *poutch*), while that for *xylophone* says that a low value should be used for Sub['x','z'].

The values used are in the range nought to five, simply because this was as wide a range as I found I needed. A correct substitution

scores zero, so the closeness score for a perfect match is zero. A likely error, such as the omission of a 'silent' consonant letter or making a double letter single (or vice-versa), scores 1; an unlikely error, such as substituting a *p* for an *x*, scores five, while errors of medium severity score two to four. The closeness scores for candidate words in the shortlist generally come out at between one and twelve. Table 8.1 shows the first five candidates in the shortlists for *sissors*, *satter* and *spaggety*, with their closeness scores:

Table 8.1 Candidates and closeness scores

SISSORS		SATTER		SPAGGETY	
scissors	2	setter	1	spaghetti	4
sissies	6	sitter	1	spigot	6
seesaws	6	satyr	3	spigots	8
saucers	6	sitar	3	sparsity	8
Caesars	6	shatter	3	spaced	9

Information about special Om and Sub values can be included in a dictionary entry by modifying the storage of individual letters – one can imagine a letter having a flag stuck in it indicating that it is to receive special treatment in the string-matching. This means that separate occurrences of the same letter can receive different treatment within a single word; the first *c* of *science*, for example, can be given a low Om value, while the second one can be given a low value for Sub['c','s']. Ins values are more awkward since information needs to be included both about the letters that may be inserted and about where they may appear. (The prototype actually contains only strings of insertable letters with no information about where they are allowable; this has the undesirable effect that, for instance, *potuch* would receive the same score as *poutch.* This is not a serious problem in practice since errors like *potuch* seem to be rare, but it could be put right without great difficulty.)

To get the information about particular letters of particular words into the dictionary, I began with a dictionary containing pronunciations as well as spellings.[4] For each word, a program generated a naive spelling from the pronunciation simply by converting each phoneme into a letter or pair of letters. For instance, the pronunciation of *ghoul* is represented in the dictionary

as /ɡu:l/, and the naive spelling generated from this would be *gool.* The program then calculated the closeness score between this naive spelling and the correct spelling. If the score was not zero, the program tried flagging the letters of the correct spelling in various ways, recalculating the closeness score each time. If it reduced the closeness score, then this flag was retained in the dictionary entry. For example, starting with the pronunciation /jɒt/ for *yacht,* the program would have produced *yot* and would then have calculated that the closeness-score for *yacht/yot* was greater than zero. It would then have discovered by trial and error that a low value for Sub['a','o'] gave a better score, as did low values for Om['c'] and Om['h'], so it would have retained this information in the entry for *yacht.*

In trying substitutions, it was restricted to a table of about thirty which I had set up in the expectation that they would be useful, including substitutions such as ['c','s'] (for words like *cereal*) and ['g','f'] (for words like *cough*). The phonemes /ə/ (the first vowel of *about*) and /ɜ:/ (as in *bird*) were given special treatment. Special flags were made available for the vowel letters meaning 'Any other vowel letter may be substituted here', and these were routinely given to vowel letters corresponding to /ə/ or /ɜ:/. The second *a* of *guarantee,* for instance, would be flagged in this way so as to anticipate *guarentee, guaruntee* and so on. Vowel letters that corresponded to no phoneme at all would be given two flags – one of these special ones and another to indicate a low Om value. This would happen to the first *a* of *separate* (adjective), to anticipate *seperate* and also *seprate.*

At an early stage of my research, I had assembled various collections of misspellings from spelling tests and free writing, some of them containing several thousand spelling errors.[5] I made use of these at this stage by running the corrector over these misspellings, listing those on which it performed badly. Very often it was possible to see patterns in these misspellings and to put information into the dictionary accordingly. One improvement was to include a low value for Ins['r'] in words with an /ɑ:/ (the vowel of *car*) in the pronunciation but no *r* in the spelling, such as *banana.* Another improvement involved words containing *m* or *n* followed by two of the letters *b,p,g,k,d,t* such as *handkerchief* and *presumptuous;* the first of the two letters was given a low Om value,

to anticipate *hankerchief* and *presumtuous.*

Many of these adjustments related to 'phonetic' misspellings, but not all misspellings are phonetic. For example, if a word has two short syllables ending with the same consonant (*remember*), people sometimes miss out the second one (*rember*); so the *em* of *remember* can be flagged as letters that should get low Om values. Even a single word can be fixed. *Lastest* is a common misspelling of *latest*, partly because of the pattern of the word but also, I suspect, because of its meaning; this word can be flagged as having a low value for Ins['s'].

All this may seem somewhat ad hoc, but that is its virtue. The main reason why people make misspellings in English is that English spelling is quirky; a corrector has to know about the quirks. This system offers a method of incorporating into a corrector large amounts of information about particular words or groups of words and about the ways in which people are likely to misspell them. An advantage of this system is that a corrector could be tailored for a particular group of users. If, for example, German or Japanese users of English made particular sorts of spelling errors, the dictionary could be adapted to anticipate precisely their kinds of errors; the program would remain the same.

Two variations

When I describe the above system to people, there are two suggestions they are inclined to make, both of which have appeared in the literature.

The first is based on the observation that, though English orthography has a reputation for being unruly, there is in fact a good deal of pattern in it. For example, the *c* of *sc* is hardly ever pronounced when followed by *e (scene, scent*) or *i (science, scissors*). Rather than flag the *c* in all these words (and more) as having a low value for Om['c'], would it not be better to encode this pattern and other such patterns in a table? This is essentially the system employed by Fawthrop, though he uses a different string-matching algorithm (Yannakoudakis and Fawthrop 1983a, Yannakoudakis and Fawthrop 1983b).

Having patterns like *sce/i* stored just once in a table rather than many times over in the dictionary is, in some sense, more economical and perhaps more elegant, but it runs into trouble with exceptions. Spurious patterns sometimes arise from the juxtaposition of morphemes, such as *shepherd* (compare *aphid*) and *hothouse* (compare *father*). Many words simply don't obey the rules – compare *sceptic* with *sceptre.* A corrector may choose simply to ignore these and hope for the best, but, if it is to take account of them, either it has to include exception flags in the dictionary or it has to make its table very complicated. A corrector that has all the information in a dictionary takes exceptions in its stride. It also has less work to do at run time, which is important since the user does not want to wait long for the corrector's list of suggestions.

A second suggestion is proposed by Veronis (1988). He notes that, in French (the same applies to English), there are often several ways of writing the same phoneme, such as *o* and *eau.* Where several letters are used for a single phoneme (like *eau*), he suggests that they be treated as a single unit rather than as separate letters. For example, if someone writes *bo* for *beau,* then the *o* for *eau* should be treated as a single substitution.

In terms of the graph traversal, this is the same as adding extra arcs to the network, as shown in Figure 8.5.

In addition to the arcs corresponding to single-letter omissions, insertions and substitutions, there is also an arc (from node H to node O) corresponding to the substitution of *o* for *eau.* The calculation of the minimum cost of reaching the final node (O) is the lowest of the following:

Cost at N + Ins['o']
Cost at M + Om['u']
Cost at L + Sub['u','o']
Cost at H + Sub['eau','o']

This modification neatly handles those cases where the speller substitutes one spelling-unit for another. An English example would be allowing *or* to count as a single substitution for *augh* in words like *caught* and *taught.*

By contrast, handling these spelling-unit substitutions by using only single-letter operations is clumsy. To take another example, *x* for *cs* is a common substitution in the word *ecstasy.* The above

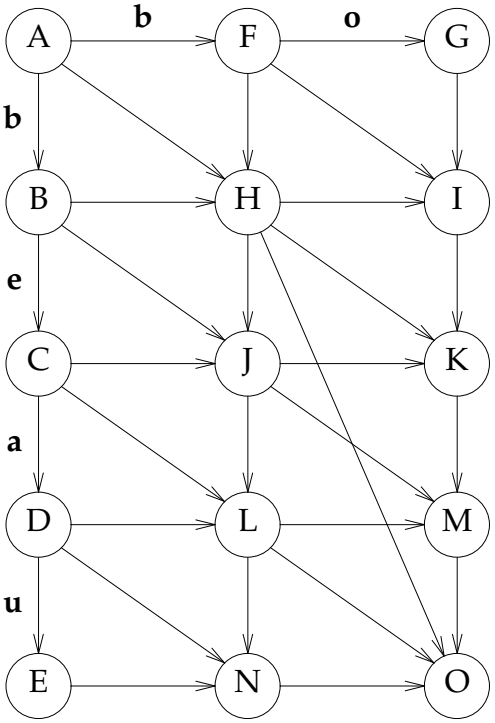


Figure 8.5

scheme would have a single substitution cost for Sub['cs','x']. Using only single-letter operations requires that both the *c* and the *s* be flagged as substitutable with *x,* that one of them (it doesn't much matter which) should be flagged as omittable, and, finally, that the *c* should be substitutable with *s* and vice-versa.

However, while the anticipation of spelling-unit substitutions would be a useful addition to the system, it would not remove the need for the single-letter arrangements. This is because poor spellers produce not only the precisely anticipated misspellings, but all sorts of variants besides. For example, *ecstasy,* as well as coming out as *extasy,* may well appear as *ecxtasy, exstasy, ectasy, estasy* and *esctasy.* While the special arc that anticipates *x* for *cs* would handle *extasy,* you would still need the single-letter system to handle all the others.

Transpositions and homophones

The prototype system as described so far has one or two weaknesses. Consider, first, how it would match *peolpe* with *people.* Having only the options of substitution, omission and insertion, it would consider the following three ways in which *pl* might have been turned into *lp:*

1. Substitute *l* for *p*, then *p* for *l.*
2. Omit *p;* substitute *l* for *l;* insert *p.*
3. Insert *l;* substitute *p* for *p;* omit *l.*

Since there are no special low-value flags assigned to any of these operations, each operation would have a cost of five, giving ten as the closeness-score, indicating a rather poor match. The trouble is that it is counting this as two errors, whereas we would feel that the writer has made just one error – getting the *p* and the *l* the wrong way round – and that *people* and *peolpe* are quite a good match.

This type of error – the transposition of adjacent letters – can be accommodated in the algorithm by having a transposition cost (transcost) defined as a constant, and introducing a short section into the program just before the end of the nested **for** loop, as follows:

```
if (i > 1) and (j > 1)
  then if (Word[i–1] = Misp[j]) and (Misp[j–1] = Word[i])
       then begin
            arctrans := Cost[i–2,j–2] + transcost;
            if arctrans < Cost[i,j]
              then Cost[i,j] := arctrans
           end
```

This has the effect of inserting an extra arc, as shown in Figure 8.6.

We can get from A to D for a cost of zero since the *peo* of the misspelling matches the *peo* of *people* exactly. Then there are three routes from D to L corresponding to the three ways described above of turning *pl* into *lp*. They are as follows:

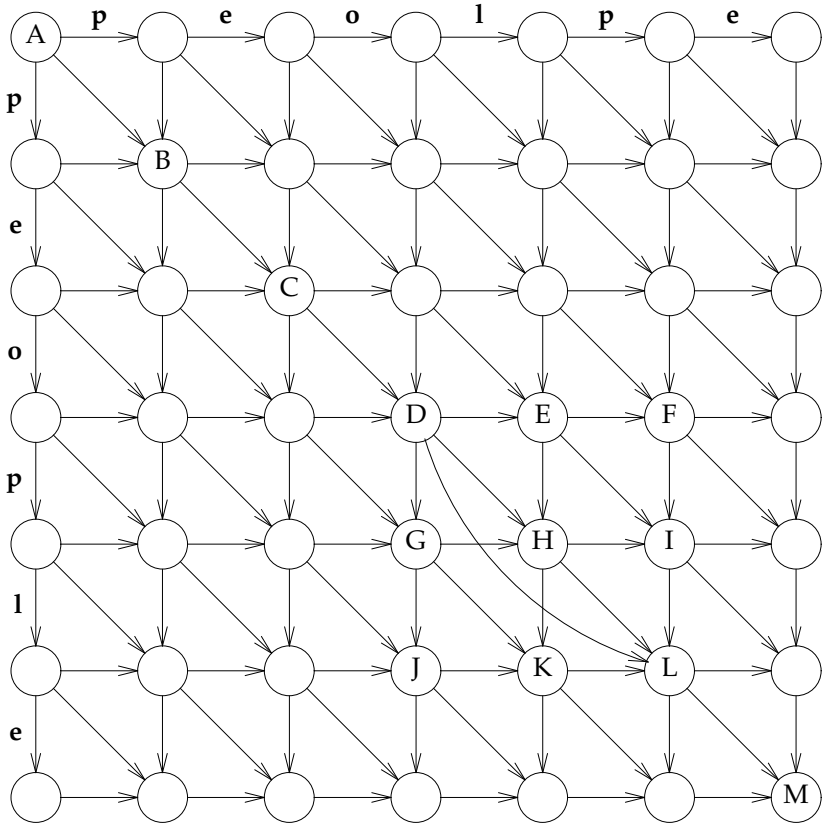


Figure 8.6

1. Substitute *l* for *p*, then *p* for *l.* D-H-L
2. Omit *p;* substitute *l* for *l;* insert *p.* D-G-K-L
3. Insert *l;* substitute *p* for *p;* omit *l.* D-E-I-L

The effect of incorporating a transposition cost into the algorithm is to add an arc direct from D to L. This cost is set as five in the prototype corrector, so we can now get to node L at a cost of five. Since the final step L-M is a zero-cost substitution, we end up with five, rather than the previous ten, as the closeness score for *people* and *peolpe*.[6]

The other weakness becomes apparent when the system is required to provide a shortlist of candidate words for a real-word error, i.e. when the misspelling is itself a dictionary word. This happens when the context check, which is explained in Chapter Ten, detects an error. For the misspelling *their* in *their_was*, for

example, the system would produce *tear*, *they*, *thee*, *other*, *there*, *theory*, *they're*, all with a closeness score of four. It would fail to put *there* at the top of the list because the string-matching algorithm does not explicitly take account of homophones.

The solution adopted in the prototype is simply to allot a code to each pair (or trio or occasionally even quartet) of homophones in the dictionary – there are about two and a half thousand of them.[7] After the string-matching has produced a closeness score, the program sees whether the candidate word is a homophone of the misspelling and then reduces the score, or not, accordingly, somewhat as follows:

```
if cand_homoph_code = missp_homoph_code
then  score := (score + 2) / 3
```

In the above example, this would have the effect of reducing the score for *there* and *they're* (both homophones of *their*) from four to two, thereby moving them to the top of the list.

Lest there should seem to be any mystery about this formula, or others that I present later, let me say now that a number of other formulas would probably do just as well. No great precision is claimed for the closeness scores themselves; their function is merely to rank the candidate words in order. Sometimes we may wish to move some of the words up or down the list, in which case a formula is chosen which moves the words around in a way that corresponds to what we wanted.

Silent corrections

At present, the corrector simply presents a list of candidates to the user, ranked in order from best guess downwards; the user does not see the closeness-scores at all. It would, however, be possible to make some more use of the scores. If, say, the first three candidates were all reasonable possibilities for some misspelling whereas all the rest were a lot less likely to be the required word, perhaps the user would find it helpful to have this indicated in some way.

Another way of using the scores would be to offer the user the option of having 'silent' corrections. Like most spellcheckers, the prototype goes through the rigmarole of offering a list of corrections and asking the user to choose, however obvious the correction. It would be possible for the corrector simply to insert its best guess in place of the error, highlighting the word in some way so that the user could check that this was in fact the word required. It would do this when the first of its suggestions had a much better (i.e. lower) score than any other.

Experiments with the test passages given in Appendix Three suggest that, when the best guess has a score less than half that of its nearest rival, it is likely to be the required word. Examples of errors which gave rise to a guess that was easily the best included *Japannese, dicided, towords, radyator, eveluation, compatable* and *posible.* Out of all the 250 errors that the prototype spotted, 77 (about thirty per cent) were of this kind. Only one of these 'easily-the-best-guess' corrections was not the word required; the error was *basced* corrected to *basked* when the required word was *based.* For writers who make a lot of errors and who resent the repetitive dialogue with the corrector, it looks as though silent correction could reduce the tedium appreciably.[8]

Notes

1. Dynamic programming was developed in the 1950s, mainly by Richard Bellman at the Rand Corporation (Bellman 1957). It was called 'dynamic' because it was devised originally for problems in which time was an important element, such as inventory control or missile guidance. It can, however, be applied, as here, to problems in which time plays no part. Similar directed networks are used in speech recognition, where they are known as Hidden Markov Models (Rabiner and Juang 1986). The algorithm I describe for finding the lowest-cost route across the network is sometimes called the Viterbi algorithm (Forney 1973).

2. This is not strictly true. Depending on the type face, an optical character reader can misread two letters as one, such as *ll* as *U,* or *rn* as *m,* especially with ligatures, but you might prefer to regard these as substitutions rather than omissions (Sun et al. 1992).

3. The network-traversal cost – which I have called the closeness score – is sometimes referred to as the 'Levenshtein distance' after a Soviet scientist who used essentially this method for error-correction of binary codes (Levenshtein 1966, Hall and Dowling 1980). It is applied to spelling correction by Wagner and Fischer (1974) who call it the 'minimum-edit distance', and also by Okuda et al. (1976) very much in the way I have presented it above.

4. This is a dictionary of spellings, pronunciations and word-classes generated from the machine-readable text of the Oxford Advanced Learner's Dictionary of Current English, third edition (Mitton 1986). It is available to researchers through the Oxford Text Archive, Oxford University Computing Service, 13 Banbury Road, Oxford OX2 6NN, U.K.; email: archive@vax.oxford.ac.uk

5. These corpora of spelling errors are also available from the Oxford Text Archive (Mitton 1985).

6. Lowrance and Wagner (1975) present an extension of the algorithm that also handles transpositions. It is a general solution covering transpositions of letters no matter how far apart they may be; for instance, it would enable *tompucer* to be considered as deriving from *computer* with just one transposition error. Such errors are rare, so I have adopted a simpler version of their extension which handles the transposition only of adjacent letters.

7. For the purposes described here, words were homophones if they had separate entries in the dictionary and their pronunciations were the same. As well as genuine homophones like *toxin* and *tocsin,* this included words that differed only in initial capitalization such as *turkey* and *Turkey,* alternative spellings such as *swap* and *swop,* spellings with and without accents such as *soiree* and *soirée,* and words ending with an unstressed *man* or *men* such as *batsman* and *batsmen.* It also included all the inflected forms – not just *ion* and *iron* but also *ions* and *irons,* not just *sew* and *sow* but also *sews* and *sows*, *sewed* and *sowed*, *sewing* and *sowing*, and *sewn* and *sown.*

8. There is an 'auto-correct' facility in version 6.0 of Microsoft Word. If you type, for example, #*teh*# (where # is some character that marks a word boundary, such as a space or a punctuation mark), the wordprocessor will change it to *the* as you begin typing the next word, and it does this so discreetly that you might not notice that it has done it. It does this for a handful of common typos and you can get it to do the same for your own habitual errors. It simply replaces one string by another; you could, if you wanted, set it up so that if you typed *moreinfo* it would replace it by *Please do not hesitate to contact me if you need more information.* The 'silent' corrections that I am suggesting are not quite so silent – the user should certainly be aware that a change had been made – and they emerge from the correction algorithm rather than from a replacement table.

CHAPTER NINE

Restricting the search

The last chapter showed how the prototype corrector, given about a hundred words from the dictionary that seem to be reasonably promising candidates for a misspelling, ranks them in order so that it can offer a shortlist to the user, with its best guess at the top. This chapter looks at how the corrector selects the reasonably promising hundred out of a dictionary of about 70,000.

I introduced the Soundex system in Chapter Seven and showed how it could be used to produce a list of candidates. Briefly, each word in the dictionary is given a Soundex code which encapsulates certain of the word's salient features. The hope is that any misspelling of a word will contain at least these same salient features, so that the misspelling and the correct spelling will have the same code. A code is calculated from the misspelling, and all the words in the dictionary that have this same code are retrieved to form the candidate list. The details of the code (Knuth 1973) are presented in Figure 9.1, with some examples.

Take as an example the misspelling *disapont.* A corrector would compute the code D215 from *disapont* and then retrieve all the words with code D215: *disband, disbands, disbanded, disbanding, disbandment, disbandments, dispense, dispenses, dispensed, dispensing, dispenser, dispensers, dispensary, dispensaries, dispensable, dispensation, dispensations, deceiving, deceivingly, despondent, despondency, despondently, disobeying, disappoint, disappoints, disappointed, disappointing, disappointedly, disappointingly, disappointment, disappointments, disavowing.*

The prototype corrector uses Soundex but not in its original form. Soundex was invented in America, as one can see from the

1) Keep the first letter (in upper case).
2) Replace these letters with hyphens: *a,e,i,o,u,y,h,w.*
3) Replace the other letters by numbers as follows:
   b,f,p,v : 1
   c,g,j,k,q,s,x,z : 2
   d,t : 3
   l : 4
   m,n : 5
   r : 6
4) Delete adjacent repeats of a number.
5) Delete the hyphens.
6) Keep the first three numbers or pad out with zeros.

For example:

Birkbeck	Zbygniewski	toy	car	lorry	bicycle
B-621-22	Z1-25---22-	T--	C-6	L-66-	B-2-24-
B-621-2	Z1-25---2-	T--	C-6	L-6-	B-2-24-
B621	Z125	T000	C600	L600	B224

Figure 9.1 The Soundex code, with some examples

prominence it gives to the letter *r*. Americans generally pronounce the phoneme /r/ even when it is at the end of a word or is followed by a consonant phoneme – *Ma* (mother) and *mar* (spoil), *lava* (volcanoes) and *larva* (insects) are not homophones for most Americans. I had British people in mind, however, in designing my spellchecker, and most people in the British Isles only pronounce /r/ when it's followed by a vowel sound – compare *far away* and *far behind*. Though there are some exceptions, notably the Scots, the Irish and people in the West Country (Wells 1982), I felt that the letter *r* was not so salient for most British people, so I included it along with *h* and *w* and so on in the list of letters to be ignored.

The second problem with Soundex, when used for spelling correction, is that it takes the first letter of the misspelling as part of the code. I noted in Chapter Four that the majority of misspellings have the same first letter as the correct spelling, but not all of them do. The candidate lists retrieved for *rong, enything* and *neumonia* would not contain *wrong, anything* and *pneumonia,* so a corrector

based on Soundex would fail to correct misspellings of this kind though they are trivial errors that ought to be easy to correct.

There are two ways of dealing with this, and both are used in the prototype. The first is to treat some words that begin with different letters as if they began with the same letter. For example, words beginning with a *j* are assigned the same code as words beginning *g,* so that, say, *jipsy* would get the same code as *gypsy*. The second is to compute a code from the pronunciation of a word as well as from its spelling, and to store the entry more than once in the dictionary file if the codes are different. For instance, *psyche* produces code P220 whereas the pronunciation produces S200, so the entry for *psyche* is stored twice.

The second method obviously enlarges the dictionary file whereas the first method does not. The second method also deals with another, related problem, which arises when the misspellings of a word are likely to have a different consonant pattern from the correct spelling, as in *dett* for *debt.* Since the code is computed from the pronunciation as well as from the spelling, the entry for *debt* will be included twice in the file, under D130 and D300.

Misspellings that are not based on pronunciation can also be anticipated in this way. The dictionary file contains information about letters likely to be omitted, substituted and so on – as explained in the last chapter – and this can be used in computing extra codes. The second *m* of *remember,* for instance, is flagged as likely to be omitted, so the code R510 is computed as well as R551.

Further codes are computed, if necessary, to make sure that a homophone has an entry under its partner's code as well as under its own. For example, if *sought* were detected as a real-word error – in 'this is the sought of thing,' for instance – then we would want *sort* to appear in the candidate list. However, the group of words with code S230 does not contain *sort.* (Group S200, the group for *sort,* contains *sought* because of the pronunciation, but not vice-versa.) So *sort* is simply added to group S230, and likewise for all pairs of homophones.

A final group of errors that has to be anticipated in this way is caused by typing slips that affect the first letter, such as *nad* for *and*. It is not feasible to attempt to anticipate all possible errors of this kind since each word would require many entries and the dictionary file would be enlarged many times over, but a corrector

ought to be able to cope with *nad, hte* and the like, so extra entries are included for about a thousand common words; *and,* for example, has code N300 as well as A530, so that it will appear in the candidate list for *nad.* If the corrector encountered an error of this kind in some other word, such as *ocmputer,* it would fail to retrieve the correct word (*computer*) in its candidate list, but such errors are rare.

The effect of entering a number of words more than once in the dictionary file in all the ways just described is to increase its size from about 70,000 entries to about 85,000.

These Soundex-like codes divide up the dictionary into a number of groups of entries, each group sharing a particular code. It would be convenient if these groups were about the same size – say 850 groups of a hundred words each. Unfortunately the pattern is quite different. Nearly two thousand groups are produced; the largest groups have a few hundred entries while there are several hundred groups with fewer than ten entries each. These small groups are a nuisance. As I will explain shortly, the corrector in the course of producing candidates for a misspelling actually retrieves not just one group but several, and, on the system on which the prototype was developed, it is more efficient to retrieve a hundred entries at one go than ten groups of ten.

The solution is simply to collapse together various collections of small groups. Having computed a code, the corrector consults a table of codes that are to be recoded, and changes the code if necessary. For example, the groups F515, F514, F513, F512, F511 are all joined with F510 to make a single group. After these rearrangements, there are about 750 groups, and the smallest group has fifty entries.

This modified Soundex system copes with the sort of errors that have some fairly obvious motivation – double letters in place of single (and vice-versa), 'silent' letters, unstressed vowels and so on. It does not cope well, however, with keying errors and with unexpected, but still correctable, spelling errors. For example, it is easy to see what *atlogether* is meant to be, but the group retrieved by the code A342 will not include *altogether,* so the corrector will fail to offer the right word.

Taking it as a target that the corrector should at least be able to cope with misspellings that have just one single-letter error

(substitution, omission, insertion or transposition) at any place in the word after the initial letter, then the corrector will have to retrieve a number of groups with codes slightly different from the misspelling's code. For example, to cope with the transposition of the leading consonants in *atlogether,* the corrector would find the right word in group A432, not the misspelling's own group (A342).

Having computed the misspelling's code and having retrieved and considered the words in that group, the corrector then generates all the related codes that could contain a word differing from the misspelling in one of the above four ways and retrieves the words in those groups. Depending on the original code, this can produce over forty codes. This need not mean that over forty groups need to be retrieved, however, for two reasons. The first is that some of these codes correspond to empty groups – there never were any entries in the dictionary with these codes. The corrector has a table of these and can simply delete them from the generated list. The second is that subgroups of these related codes often appear in the table of codes that are to be recoded, so several of them get collapsed together to form larger groups.

Even so, this procedure increases substantially the number of words to be considered, and a large proportion of these words bear very little resemblance to the misspelling. To save time, the corrector performs a quick test on them. It arranges the letters of the misspelling into letter-string form (that is to say, in alphabetical order and without repeats); *mutliply,* for instance, would become *ilmptuy.* Each dictionary entry contains its spelling in both original and letter-string form. Having retrieved an entry, the corrector compares its letter-string with that of the misspelling. If the two differ by no more than a certain number of letters (this number is larger for longer strings), the candidate has passed this quick test and is passed on for the full string-matching as described in the previous chapter. This quick test winnows out most of the entries from these related-code groups.

One final, and important, way of restricting the search is to take account of word length. People's misspellings are generally about the same length as the words they are trying to spell. Table 9.1 shows how misspellings from a spelling test differed in length from the correct spellings.

Table 9.1 Lengths of misspellings from a spelling test

		Missps of all words	*Missps of short wds (3 to 5 letters)*	*Missps of long wds (10 letters or more)*
Missps were:	–3 or more	6%	0%	11%
	–2	12%	1%	14%
	–1	40%	26%	41%
	same length	32%	49%	27%
	+1	9%	20%	6%
	+2	1%	2%	1%
	+3 or more	*	2%	**
Number of missps (=100%)		2514	96	1194

* 8 missps (= 0.3%) ** 2 missps (= 0.2%)

The table shows, for example, that, out of all 2514 misspellings, thirty-two per cent were the same length as the words they were misspellings of. Forty per cent were one letter shorter, twelve per cent were two letters shorter, while six per cent were three or more letters shorter than the words they were misspellings of, e.g. *betful* for *beautiful.*

These misspellings were taken from a test given to fifteen-year-olds in Cambridge in 1970 – the same people who produced the corpus of errors described in Chapter Four. They are typical of spelling test results. The majority of misspellings are close to the length of the correct spellings. Misspellings of short words often err on the side of being longer than the correct spellings (depending somewhat on the actual words used in the test), while misspellings of long words tend to be shorter than the correct spellings. Comparison of results from a number of tests indicates that this bias is more marked in the misspellings of poorer spellers.

Spellcheckers are intended to correct the misspellings that people make in free writing, not in spelling tests, and it is possible that the errors they make in free writing follow a different pattern. The following table presents the same analysis as the previous one, but this time based on misspellings from free writing, in fact from compositions written by the same people who did the spelling test.

Table 9.2 Lengths of misspellings from free writing

		Missps of all words	*Missps of short wds (3 to 5 letters)*	*Missps of long wds (10 letters or more)*
Missps were:	–3 or more	1%	*	2%
	–2	9%	3%	14%
	–1	38%	37%	40%
	same length	30%	33%	31%
	+1	20%	24%	12%
	+2	2%	3%	1%
	+3 or more	**	0%	0%
Number of missps (=100%)		3351	1145	349

* 1 missp (= 0.09%) ** 1 missp (= 0.03%)

Misspellings of hyphenated words and misspellings containing spaces, such as *head miss dress* for *headmistress,* are not included in this table.

This table differs somewhat from the previous one, and this difference is good news for spellcheckers. In the spelling test results of Table 9.1, eleven per cent of the misspellings of long words were three or more letters shorter than the correct spellings; the corresponding figure from Table 9.2 is only two per cent. All but a handful of the misspellings in free writing were within two letters (in length) of the correct spelling. The following are examples of the few misspellings that differed by more than two letters: *parallellel, hankichies, teral (terrible), divent (different), pertickly, semblys (assemblies), seticates (certificates).*

The difference between the two tables has a simple explanation: poor spellers use short words. In a spelling test, everyone has to attempt all the words. The poor spellers get the longer (usually harder) words wrong, sometimes very wrong, and often produce misspellings that are much shorter than the correct spellings. They know they cannot spell these words so, in their own writing, they simply don't attempt them. It is not that these words are outside their vocabulary, but that they curtail their vocabulary, perhaps severely, in an attempt to reduce the number of spelling errors

(Moseley 1989). The misspellings of long words in Table 9.2 are largely the work of the better spellers, whose misspellings are usually close, in length and in other ways, to the correct spelling.

There are, of course, exceptions to this. A poor speller occasionally has to attempt a long word, as in the *semblys* and *seticates* above. And there are some poor spellers who regularly use long words and spell them badly (Frith 1980), but they are exceptions. In the corpus from which the above results were taken, the following passages are typical, the first of a poor speller, the second of a good one:

> In my primary shcool, it was roton because you we treated like littal babys, And you had to do babys work, and that was no good, and if you did'ent do ore home work you were smacked on the legs one day when the teachur smacked me on the legs, I put my tun out to her, so she done it agian, my last teachur at my old school was Mr Woods and he was a nice teachur and we got on with him we done owe work good and so for that he used to take us on visits

> We did Maths and English perhaps for two hours at a time, with the occasional woodwork and extra games period. I remember also that I used to be quite good at everything including cricket and football which I certainly am not now. I was given the appointment of Head Boy which seemed a bit ridiculous because it made no difference to myself and anybody else in the school as I wasn't really made a prefect and had no authority.

Since a corrector is meant to cope with free writing rather than with the results of spelling tests, it can restrict its search to words that are just one or two letters longer or shorter than the misspelling, perhaps taking in a slightly larger range on the long side for longer misspellings, to include long words that may have been shortened.

The result of restricting the search in all these ways is that the corrector performs the string-matching on a relatively small number of candidates. The actual number varies from one misspelling to another – for example, it takes 155 candidates for *sissors* and 90 for *mutliply* – but the selection procedure is pretty reliable. If the required word is in the dictionary at all, it is almost always in the list that gets considered by the string-matching.

CHAPTER TEN

Using context and other information

As described so far, the corrector makes no use of context. It retrieves a set of candidates simply on the basis of certain features in the misspelling and then it puts them into order by matching them against the misspelling, one by one. It would work just as well on a string of unconnected words as on meaningful prose. But context is often useful, and sometimes essential, for detecting and correcting misspellings.

Human readers have a great advantage here over computers: they understand the text. They know what the words mean and what the writer is trying to say. They may also know a lot about the writer and about the situation that the text refers to. When correcting a misspelling, they can bring all this knowledge to bear in deciding what word the writer intended.

The prototype corrector does not understand the text it is trying to correct, but it does make use of context in a small way. Its dictionary contains information about parts-of-speech – noun, verb and so on – and it uses this information both to detect errors and to improve its list of suggestions.

A project at Lancaster University, mentioned in Chapter Seven, has produced a computer-readable corpus of one million words of English text, each word accompanied by its tag, i.e. its part of speech.[1] For example, the phrase 'after a lifetime of healing the sick' is tagged with codes indicating 'after *(preposition)* a *(indefinite article)* lifetime *(singular noun)* of *(preposition)* healing *(present participle or gerund)* the *(definite article)* sick *(adjective)'*. It is easy to take this tagged corpus and to generate a table showing how often each tag was followed by each other tag. The definite article, for

example, was frequently followed by an adjective or a noun but almost never by a personal pronoun or a verb. The corrector uses this table in two ways: first for trying to pick up real-word errors, later for improving the order of the candidate-words in its shortlist.

In trying to spot real-word errors, the corrector just makes use of the zero entries in the table. Suppose it is checking the phrase 'this is the sought of thing ..'. It knows, from the dictionary, that *the* is a definite article and that *sought* is the past tense of a verb, and it finds, on consulting the table, that this combination (definite article followed by past tense of verb) never occurred in the corpus. It therefore flags the pair *the_sought* on the grounds that *the* or *sought* is likely to be an error.

Of course, this procedure often fails. It might flag a combination of tags that is unusual but valid; that a particular combination failed to occur in a million words does not necessarily mean that it is unacceptable English. More often, it will fail to flag an error if there is some interpretation of the words that makes it syntactically possible. For example, 'the goes of a chance' at first sight looks as though it would be flagged because *goes* is a verb, but *goes* can also be a noun ('you get three goes for sixpence'), so the corrector would fail to flag it. I have not conducted extensive tests, but my impression is that it detects about ten per cent of real-word errors.

In its favour, the procedure is quick and simple. It picks up a few of the real-word errors that dictionary look-up cannot spot, and it rarely generates a false alarm – an acceptable combination of tags that never occurred in the corpus is certainly unusual. It also picks up some errors other than spelling errors, in particular words that have been repeated or omitted. 'Paris in the the spring' would be flagged (*?the_the),* as would 'to suffer the and arrows of outrageous fortune' (*?the_and).*

The table used by the corrector is not, in fact, exactly the one you would get by generating it straight from the corpus in the way I suggested above. The corpus uses 153 tags; some of these occur very rarely, and others have no counterpart in my dictionary. By combining some tags together, ignoring some others, and creating some new ones in my dictionary, I arrived at a set of eighty-eight that I could use in the table. (This is not as drastic a reduction as it may seem since the great majority of words get the same tags in both systems – singular nouns, base form of verbs etc.)

When creating the table, I ignored headlines and phrases containing cited words or foreign words since their syntax differs from that of ordinary prose, and I also modified the generated table slightly, on the one hand to pick up more errors and, on the other, to suppress some false alarms. Many tag pairs occur only once in the corpus, and I inspected all of these in their context. A few turned out to be errors (i.e. they were errors in the original documents, faithfully retained in the corpus), so I changed these to zeros in the table; there is one instance of *the the,* for example, and it is quite clear from the context that it was a mistake. Similarly, some pairs that were pretty odd, either because the writer was representing ungrammatical English ('we was just getting down to business'), or because he was using a word in a strange way ('the yourself is thy greatest enemy') were also changed to zeros. On the other hand, some tag pairs have zero occurrences simply because each of the two tags is relatively rare in its own right, but the pair is in fact acceptable, so I changed the zero value in these cases to a code meaning 'rare but acceptable'. I simply inspected all the table entries (rows and columns) for tags occurring less than a thousand times in the corpus. For example, some tags are used for only one word; *both* and *did* are among these, and the phrase 'both did' does not occur in the corpus but it is obviously acceptable English.

The syntax checking is in fact slightly more complicated than my earlier brief description suggested. Suppose that the first of three words has tag A, the second has two tags – B1 and B2 – and the third has tag C. It is possible that A followed by B1 is acceptable but not A followed by B2, whereas B2 followed by C is acceptable but not B1 followed by C. So, although the first word can be followed by the second, and the second word can be followed by the third, there is no tag sequence that can take us from the first word to the third, and the corrector will flag this as a possible error. An example would be *has pluck him. Pluck* can be a noun or a verb. *Has pluck (noun)* is acceptable and *pluck (verb) him* is acceptable but *has pluck him* is not. In general, between any two single-tag words X and Y, the corrector will flag an error if there is no sequence of acceptable tag pairs going all the way from X's tag to Y's tag.

It is possible that the corrector would have more success in detecting errors if it used tag triples rather than just tag pairs. For example, while the sequences verb-noun and noun-verb are

obviously acceptable, it may be that the sequence verb-noun-verb is not. Unfortunately, the million-word corpus is not large enough to provide a reliable table of tag triples; too many of the entries would be zero just because they did not happen to occur in that corpus, not because they are unacceptable in general. To avoid this problem, I combined the eighty-eight tags into twenty-four groups and derived a 24-by-24-by-24 table of tag-triple frequencies from the corpus. The corrector uses this table in much the same way as it uses the table of tag pairs. Disappointingly, it makes very little difference; the tag-triple check picks up hardly any errors that the tag-pair check misses. On the other hand, it does not produce false alarms either, so its contribution, albeit small, is to the good. I don't know whether a larger triple table derived from a much larger corpus would produce a significant improvement.[2]

Using syntax to reorder the candidate list

As well as being used to detect a few real-word errors, the table of tag pairs is also used to reorder the shortlist of candidate words for a misspelling. Suppose the user has asked for suggestions for *sought* in 'the sought of'. Whatever the right word is, it is almost certain to be a noun, so any candidate words that can only be verbs should be moved down the list. The shortlist produced by the string-matching is *soughed, sort, sight, sough, soughs, sighed, searched. Soughed* ought to be moved down the list, purely on the grounds that it is a verb (the past tense of *sough,* meaning to make a murmuring noise, as of trees in a wind).

The corrector therefore computes another score for each candidate. This score represents how well the candidate fits into this place syntactically. As with the closeness score from the string-matching, a low score represents a good fit. For this purpose, the table of tag-pair frequencies described above is reduced to a table of single digit codes, with the following meanings:

9 zero occurrences in the corpus
7 less than 0.1% (very rare)
4 less than 1.0% (quite rare)
1 1.0% or more

The percentages need a word of explanation. Suppose there are thirteen occurrences of tag A followed by tag B in the corpus, and suppose there are 600 occurrences of tag A altogether and 140,000 of tag B. The code given to the A-B pair depends, obviously, on whether you calculate 13/600 or 13/140,000. I decided that a pair should be considered rare only if both calculations produced a low percentage. In other words, the percentages are calculated out of the lower of the two tag totals.

Suppose we have a candidate with tag Y; the previous word has tag X and the following word has tag Z. The corrector consults the table to retrieve the code for X-Y and the code for Y-Z and then computes the (syntactic) goodness-of-fit score simply by multiplying them together. For example, suppose that the candidate *gravelly,* which is an adjective, is being considered for the misspelling *gravly* in *the gravly voice.* A definite article is commonly followed by an adjective, so the code for this tag-pair is 1, and an adjective is commonly followed by a singular noun, so the code for this tag-pair is also 1; the goodness-of-fit score for *gravelly* will be 1×1 = 1. By contrast, when the candidate *gravely,* which is an adverb, is considered, the table has the code 4 (meaning 'quite rare') for definite article followed by adverb, and 4 also for adverb followed by singular noun, so the goodness-of-fit score for *gravely* will be 4×4 = 16.

If the previous word or the candidate word or the next word has more than one tag, more than one score is calculated, and the lowest is retained. For example, if the corrector were considering *orange* as a candidate for *oringe* in *a light oringe skirt,* the preceding word (*light*) would have three tags (noun, verb, adjective), the candidate would have two (noun and adjective) and the following word (*skirt*) would have two (noun and verb). The corrector would calculate twelve scores, one for noun-noun-noun, one for noun-noun-verb, one for noun-adjective-noun, and so on. It would keep the lowest (i.e. the best) as the goodness-of-fit score for *orange.*

If the word to left or right of the misspelling is another misspelling, or is under suspicion, the corrector does not attempt to make use of the suspect word when reordering the list.

The closeness score (from the string-matching) and the goodness-of-fit score (from the procedure just described) are simply multiplied together to produce a new score for each candidate, and the candidates are then reordered on the basis of these new scores. Candidates now scoring ninety-nine or over are dropped from the list. The majority of candidates are either nouns or verbs, and the effect of this procedure is, generally, to move the right sort of word to the top of the list. The result, in the case of 'the sought of', is to replace the list *soughed, sort, sight, sough, soughs, sighed, searched* with the new list *sort, sight, sighed, searched, soughed.*

Word frequency and recency

Careful comparison of the two lists just presented will show that *soughed* has been demoted not merely below the nouns but below two other verbs. The reason for this is that another factor has entered into the calculations, namely word frequency.

The dictionary contains a number of decidedly obscure words, such as *haulm, repp* and *sough.* One approach, adopted by several spellcheckers, is simply to delete such words from the dictionary. I have preferred to keep them in since, after all, people do occasionally use obscure words. Besides, once you start removing words because they are obscure, it is not clear where you should stop. However, it does not seem right that, say, *home* and *haulm* should be competing on equal terms as candidates for a given misspelling; the corrector should take into account that *haulm,* purely on the grounds of frequency, is less likely to be the word required.

I gave every word in the dictionary a frequency code – very common, ordinary, or rare. The very common were those words that appeared in the five hundred most frequent words of several different frequency lists, i.e. a word had to appear in all the lists to be included.[3] The rare were those that I thought were rare, with my own estimates combined to some extent with those of two friends

of mine. I realize that this definition of rarity seems highly unscientific, but, at the time when I needed these estimates, there seemed to be no appreciably better way of producing them. I could perhaps have taken the opinions of many more people, but this would have been a long job and I doubt if the resulting list would have been much different. The problem is that computer-readable text corpora of a million or several million words, while certainly large enough to provide data about common words, are nowhere near large enough to provide data about rare words. A word that fails to appear in a corpus of, say, five million words, is not necessarily rare; conversely, a word that appears several times might still be rare in general use. In the absence of estimates drawn from hard data, it was better for the corrector to be provided with my estimates of word-rarity than with none at all.[4]

Strictly speaking, it is word-tags rather than words that have these frequency codes. *Crab,* for example, is given the code for ordinary as a noun but rare as a verb.

In order to play their part in the calculations, these codes are given number form, as follows:

1 very common
2 ordinary
6 rare

The full formula for the computation of the goodness-of-fit score is as follows:

goodness-of-fit := (XYcode * YZcode * Xfreq * Yfreq * Zfreq) / 3 + 1

As I said in presenting an earlier formula, there is no deep mystery buried in these numbers or formulas. Other numbers and formulas would probably serve as well or possibly better; in fact, finding the best way of combining these different sorts of information is a topic for further research. The straightforward methods I have used simply provide a means by which the information available about the candidates can be brought to bear upon the ordering of the list.

In my description, early in Chapter Eight, of how the corrector appears to the user, I mentioned that the corrector has two different levels of query. It looks up every word of the text in its dictionary. If it encounters a word that is not in its dictionary (except for words beginning with capital letters, about which more

later), it queries the word strongly; such words are very likely to be errors. But its other methods of detecting errors are more prone to produce false alarms, so it raises a more tentative query over errors detected in these other ways. One of these is the syntactic method described above, which enables it to catch 'the sought of' and the like. Another deals with rare words.

The problem here is that, if the dictionary contains rare words, some misspellings will match them and therefore fail to be spotted. *Wether* is a notorious example of this. Arguably, an occurrence of *wether* in a passage is more likely to be a misspelling of *weather* or *whether* than an intentional use of *wether.* If you have *wether* in the dictionary, you fail to flag these misspellings; if you don't, you incorrectly flag, say, *the wether ewe* as a non-word error. The policy adopted by the corrector is to flag (tentatively) any rare words (i.e. words for which all of the tags are rare). It is as though the corrector is saying, 'This may be OK but it looks strange; do you mind checking it?'

Recency of use, as well as frequency in general language, is taken into account when ordering the candidates. *Pedal* and *peddle* might both be strong candidates for *peddal,* but *pedal* would be preferred if *pedal* had been used earlier in the passage (perhaps having been selected by the user from lists of candidates for earlier misspellings). This is achieved by storing the most recent few hundred words in memory, along with the number of times they have occurred. Each candidate is looked up in this list and promoted up the shortlist of candidates if it is found there; the more often the word has occurred, the greater the promotion.

These few hundred words are held in a well-known data structure called a 'binary search tree'. A tree formed from the first few words of the last sentence would look like Figure 10.1.

To look up a word in a binary search tree, we begin by comparing it with the word at the top. If this is the word we are looking for, then we have found it and we finish the search. If it isn't, then we see which comes first in alphabetical order. If the one we are looking for comes first, we follow the *left* arrow down to the next level; if it's the other one that comes first, we follow the *right* arrow. We carry on like this until either we have found the word we are looking for or we get to the bottom of the tree, in which case we know that the word we are looking for is not there.

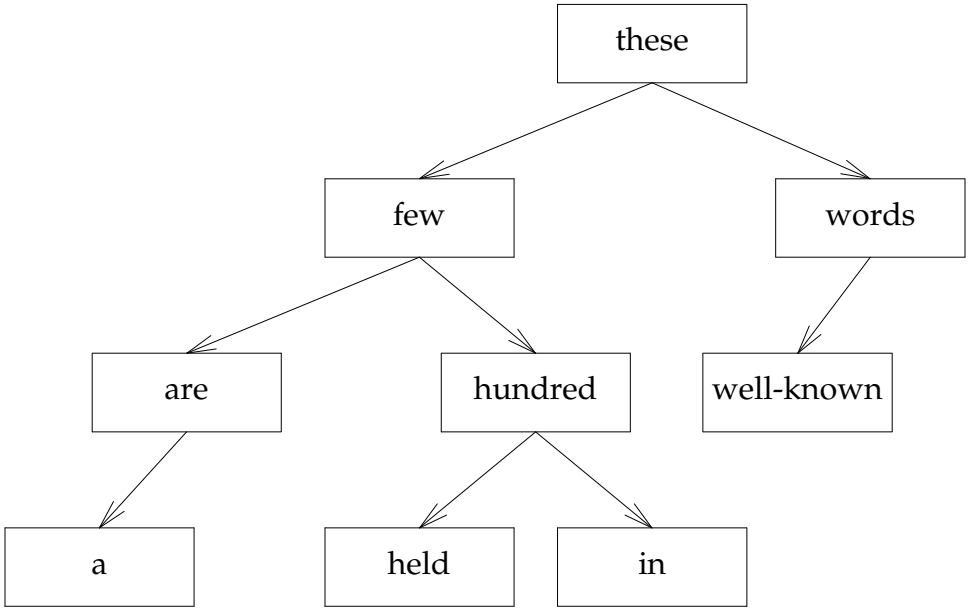


Figure 10.1

For example, if we were now looking to see whether the word *hundred* was in the tree, we would begin at the top by comparing *hundred* with *these. Hundred* comes before *these* in the alphabet, so we follow the left arrow down and compare *hundred* with *few. Hundred* comes after *few* in the alphabet, so we follow the right arrow down. This time we are lucky and we find the word we are looking for. If *hundred* was not in the tree, we would eventually reach the bottom of the tree and we would know that *hundred* was not there.[5]

Syllables

When asked to supply suggestions for *gauratee* (a misspelling of *guarantee),* the string-matching algorithm described in Chapter Eight produces *garter, grater, garotte, Gertie, garret, greater, gaiter, grittier, guarantee* (with closeness-scores of 6,6,6,7,7,7,8,8,8 respectively). A striking thing about this list, apart from the disappointing position of *guarantee,* is that only the last two

offerings have the same number of syllables as the misspelling. That this should sometimes happen is not surprising since the algorithm proceeds left to right through each word, letter by letter, taking no notice of syllable structure. Yet misspellings generally have the same number of syllables as the intended words, so the corrector does some extra work on the candidates to take account of this.

First, the corrector needs to know how many syllables each candidate has. This was easy to add to the dictionary since, for the great majority of words, a fairly simple algorithm can compute the number of syllables from the pronunciation, which was part of the information already in the dictionary entries. The exceptions, which I dealt with piecemeal, were words that seemed to have subtly varying pronunciations suggesting different numbers of syllables. *Mayor, tour* and *flower,* for instance, seem to hover between one and two syllables, while *onion* and *labelling* seem to be between two and three. Two does not seem quite enough for *aspire,* but, if you plump for three, do you have to give four to *aspiring*? I fear my decisions in such cases will have been somewhat arbitrary.

The corrector also needs to make an estimate of the number of syllables in the misspelling, and it is more difficult to estimate syllables given only the spelling (even a correct spelling, let alone a misspelling). It is not sufficient just to count vowels or groups of vowels; consider the effect of *s* or *d* on a word ending with *e,* as in *plates, places, plated, placed.* (Since we are here analysing spellings, the words 'vowel' and 'consonant' in this section refer to letters.) Prefixes and suffixes need special handling – *coin, coincide, egoist, joist.* Some consonants are syllabic, like the *m* in *prism* (though *prismatic* has only three syllables, not four). Probably the most troublesome of all is an *e* somewhere in the middle of a word – *caveman, covenant, placemat, placebo.*

The module that performs this task is quite long, because of the number of word patterns it has to deal with, and also messy because of the number of exceptions. An example of a pattern with an exception is provided by the ending *ed.* This is generally the past-tense morpheme and should be stripped, adding to the syllable count if preceded by a *t* or a *d* (*hunted, hoarded*) but not otherwise (*hoped, honed* etc). The exception arises with words

ending *bed* since, although the above applies to most of these words (*cubed, fibbed, combed, barbed* – one syllable each), it does not apply to words that have been formed by tacking the word *bed* onto the end of another (*flowerbed, sickbed, seabed* etc).

Briefly, the module first strips *un* and *under* and then final *s.* It modifies the *u* of *qu* to prevent the later parts treating it as a vowel, and likewise the *u* of *gu* when followed by a vowel (*guard, guinea).* It strips a number of suffixes and final *e* and then strips a number of prefixes. It finally analyses what remains of the word using a network of arcs and nodes somewhat like those in Chapter Eight.[6] It is best explained by an example.

Figure 10.2 is a simplified diagram of part of the network. (The network actually used by the corrector is larger and more complicated, but this will do for illustration.)

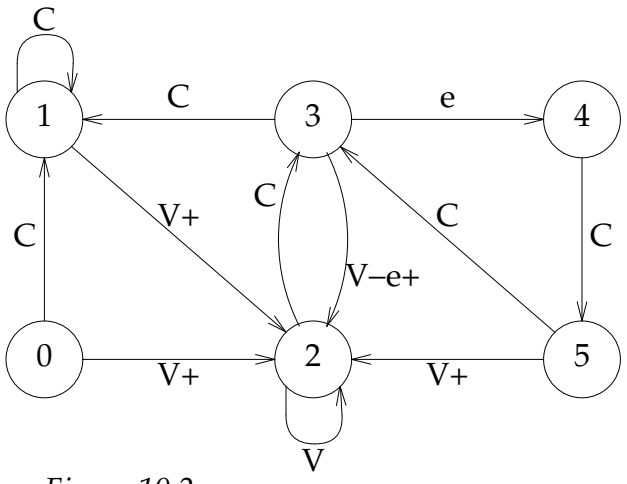


Figure 10.2

The computer begins at node nought, sets the syllable count to zero and considers the first letter. A *C* in the diagram stands for *Consonant*, a *V* for *Vowel*, and a plus-sign means, 'Add 1 to the syllable count.' If the first letter is a consonant, the computer goes to node one; if it's a vowel, it goes to node two and adds one to the syllable count. Then it takes the next letter, and so on till the end of the word. (If it were at node three, a consonant would take it to node one, a letter *e* would take it to node four, any other vowel to node two.)

For example, suppose it is analysing the word *scapegoat*. Beginning at node nought, the first letter is *s*, a consonant, so it goes to node one. The next letter is *c*, also a consonant, so it stays at node one. Next comes *a*, a vowel, so it goes to node two and adds one to the syllable count. The *p* takes it to node three, the *e* to node four, the *g* to node five and so on, as follows:

```
0   1   1 * 2   3   4   5 * 2   2   3
  s   c   a   p   e   g   o   a   t
```

Two arcs (one to two and five to two) have added to the syllable count, making two syllables for the word.

The network presented in Figure 10.2 would give the wrong result for many words. The version actually used by the corrector has ten nodes and there are many extra conditions attached to the arcs. For example, the letter *y* is counted as a consonant at node zero but generally as a vowel thereafter. Another example concerns node three. The general rule here is that an *e* takes it to node four, as in Figure 10.2, but there are some exceptions. If the letter before last was also an *e*, then it takes the arc back to node two, adding one to the syllable count (as in *event* and *television*, for instance), except where the last letter but two was an *e* or an *i* (as in *cheesecake* and *piecemeal*), in which case it goes to node four.

In the case of *scapegoat*, the whole word is analysed by the network, but usually it is only the remainder of a word that receives this treatment, after prefixes and suffixes have been removed. The examples on the next page illustrate the effects of the preliminary affix-stripping.

To test the syllable-counting module, I ran it over the dictionary, estimating the number of syllables from the spelling alone and then comparing this number with the actual number of syllables already stored with each word. It got about ninety-six per cent of them correct. To improve it further would have meant including a table of exceptions (*cafe* and the like), and there was no point in doing this since the purpose of the module was to estimate the number of syllables in misspellings, which, of course, would never be in the table. (The module is used only for non-word errors; if the misspelling is a real-word error – the *sort* for *sought* type – the number of syllables in the misspelling is taken from the dictionary entry.)

coincides	Strip final *s*
coincide	Strip final *e*
coincid	Strip prefix *co* and add 1
incid	Analysis with the network adds 2
	Total: three syllables
unambitiously	Strip prefix *un* and add 1
ambitiously	Strip suffix *ly* and add 1
ambitious	Strip suffix *ious* and add 1
ambit	Analysis with the network adds 2
	Total: five syllables
undermanagers	Strip prefix *under* and add 2
managers	Strip final *s*
manager	Strip suffix *er* and add 1
manag	Analysis with the network adds 2
	Total: five syllables

This module produces an estimate of the number of syllables in the misspelling. The shortlisting procedure, after computing a closeness-score for each candidate, increases the scores of those candidates with a different number of syllables from the misspelling, i.e. they are considered to be a less good match than those with the same number of syllables.

Divided words, function words and common typing errors

The problem of divided words was mentioned in Chapter Four – errors such as *in side* and *my self.* The corrector's approach to this problem is very simple. It has a list of words – only about thirty of them – that are prone to being split in this way. If it encounters a word which is the right-hand part of one of these (*side, self),* it looks to see whether the preceding word is the left-hand part. If it is, it queries the pair (*?in_side).*

It would be possible to widen the scope of this procedure considerably by taking every pair of words, joining them up, looking up the joined-up version in the dictionary and querying

the divided pair in those cases where the joined-up version was in the dictionary and, furthermore, had a tag acceptable in the context. This would be a lot of effort for a fairly small return, but my main reason for not incorporating it is a nervousness that its occasional successes ('He was arrested by a *?police_man*') might be more than offset by silly queries ('They live in the *?green_house* down the road').

The opposite error – of joining up words that ought to be separate, as in *alot* and *anymore* – is rare and is largely restricted to a few set phrases. It is easily handled simply by including the set phrases in full (*a lot, any more*) in the dictionary; if the user asks for suggestions for, say, *alot,* then *a lot* will be among them.

A spellchecker that was more interested in typing errors than in misspellings would have to do more than this. A system developed at Cambridge University, for example, will insert or remove spaces, as well as other letters, in an effort to correct an error. If given *witha,* for instance, it tries *with a*; if given *nev er,* it tries *never.* It sometimes generates several possibilities. For the cryptic string *th m n worked,* it tries a number of variations including *them an worked, them no worked, to man worked* and others. It passes all of these on to the next stage at which syntactic and semantic processing eliminates all except *the man worked* and *the men worked* (Carter 1992).

Another type of error I mentioned in earlier chapters is the writing of one function word for another – *the* for *there, an* for *and* and the like. These slips are common. It is not easy to spot them, though the syntactic-context method described above spots some of them, nor is it easy to provide the required candidate. Sometimes the required word will not be retrieved from the dictionary at all, such as *be* for *we;* more often, the required word, though appearing in the shortlist, will be further down the list than it should be.

The corrector deals with these simply by having a table of function words (about sixty of them) and, for each one, a short list of words that are likely candidates, with preset closeness-scores. I derived these lists of likely candidates from three collections of misspellings that included errors of this type.[7] If the misspelling is in this table, the corrector adds this list of candidates, with their closeness-scores, to the shortlist it has already selected. The preset closeness-scores are set sufficiently low that these candidates

appear at the top or close to the top of the shortlist.

The same problem arises with typing errors of common words, such as *hte* for *the* and *adn* for *and,* and the same solution is adopted. The misspelling *hte,* for instance, is included in the table with *the* as its candidate; the score given for *the* guarantees that it appears at the top of the list of candidates. The creators of the SPEEDCOP system (mentioned in Chapters Six and Seven) had much the same idea in building a dictionary of 256 common misspellings; the criterion for inclusion of a misspelling was that it occurred frequently and, when it did occur, was almost always a misspelling of the same word. With this small dictionary they found they could correct about ten per cent of misspellings, with almost one hundred per cent accuracy (Pollock and Zamora 1984).

Capitals, apostrophes and hyphens

I have now almost completed my general description of how the corrector works – how it spots misspellings, how it retrieves sets of possible candidate spellings from the dictionary and how it puts these into an ordered shortlist. I round off the chapter with a discussion of certain troublesome details in the input text.

The corrector works its way through a text, sentence by sentence and word by word. I have been assuming that there is no problem about this, but taking a text 'word by word' is not as simple as it may at first appear. Consider the sentence, 'Sugar mice and other such cavity-forming treats are bad for children's teeth.' The word *sugar* is in the dictionary with a small *s* (i.e. not *Sugar);* the words *cavity* and *forming* are in, but not *cavity-forming;* and *children* is in but not *children's.* Even assuming that the text consists of prose and not, say, names and addresses or modern poetry, there are various such matters to be attended to. The prototype corrector does not handle these as carefully as a piece of commercial software would have to, but it handles most cases acceptably.

Generally, a string of characters bounded by spaces or punctuation marks is taken to be a word, and certain punctuation marks – *.?!;* – are taken as marking the end of a sentence, possibly followed by a closing quotation mark. The main problem here is

that full-stops are often used as part of an abbreviation – *Mr.*, *i.e.*, *B.B.C.* and the like. When it encounters a full-stop, the corrector checks whether the preceding 'word' was a single letter or one of a handful of common abbreviations such as *Mrs* and *Dr.* If it was, the corrector takes the full-stop to be part of the word and not an end-of-sentence marker. (If the full-stop also marked the end of a sentence, the corrector would make the mistake of taking two sentences together as though they were one.) The corrector also copes with strings of punctuation marks such as *...* and *??!* [8]

A word that begins with a capital letter is looked up in its capitalized form since the dictionary contains a number of proper nouns that are written only with capitals (*Susan, Lancaster, Peru);* it can also happen that the capitalized form is a different word from the lower-case form (*March/march, May/may).* It is also looked up in a list of names that have already been accepted by the user. If the word is not found, and it is the first word in the sentence, it is looked up again with the capital changed to lower-case. If it is still not found, it is classed provisionally as a proper noun and queried tentatively. If accepted by the user, it is added to the list of accepted names, so that further occurrences of the same name will not be queried.

Strings of numbers, and 'words' containing numbers, are accepted without demur. It seems pointless to query *1988, 24-pin, DB2* and the like.

Apostrophes are a nuisance, especially *'s.* A small amount of extra work is required in looking words up. A word is first looked up with the *'s* retained because there are some words in the dictionary that are spelt with *'s,* such as *let's,* and then, if it is not found, it is looked up with the *'s* stripped off. But then more work is required in adjusting the tags. For some words, *'s* can only indicate the genitive (*men's, boss's);* for others, it can only indicate *is* or *has* (*all's well);* and for yet others, it can indicate either (*the train's wheels, the train's late, the train's gone).* Normally the checker and the candidate-reordering module use the tags of words in the text just as they come from the dictionary, but, in the case of *train's,* the tags of *train* (noun and verb) have to be changed to genitive noun, noun-plus-IS, and noun-plus-HAS (*is* and *has* being single-word tags).

The apostrophe can sometimes prove useful, however, since inspection of a sample of written work suggests that, though people make many mistakes over apostrophes, they rarely append them to words that cannot take apostrophes, such as adjectives. This makes it possible to spot a few real-word errors, such as 'I trod on a bare's tail,' and to prune the candidate list for misspellings that have apostrophes. The corrector does not, however, detect superfluous apostrophes as in *jobs for the boy's.*

Finally, hyphens. A '-' followed by a space is taken to be the end of a word; if it is also preceded by a space, it is taken to be a dash; otherwise, it's a hyphen. The prototype does not attempt to cope with end-of-line hyphenation. Many hyphenated words are created by joining separate dictionary words (or quasi-word prefixes, like *thermo),* such as *grant-aided* and *non-tax-deductible.* But in some cases, the parts are not words in their own right, such as *laissez-faire* or *hurly-burly.* (Or, if they are, it is accidental; the words they happen to match are quite unrelated, like *burly).*

A hyphenated word is first looked up in full. If not found, it is broken into parts and the parts are looked up. If all the parts are found, the word is accepted; otherwise, it is queried. This means that the corrector will fail to flag errors if all the parts happen to match dictionary words, such as *lazy-fair* or *hourly-burly,* but the alternative would be to flag any word not included in full in the dictionary, and this would be unacceptable since words such as *computer-designed, radio-controlled, capital-intensive* and the like are so common.

Offering corrections for a misspelled hyphenated word presents a problem. Suppose the user has written *computer-desined.* There is no point in the corrector looking for matches for *computer-desined;* it will only offer the right word by concentrating on the *desined.* But the corrector does not know that this is a two-part word; it could be a misspelling of one of the one-piece hyphenated words like *hurly-burly,* part of which happens to match a dictionary word. The rather clumsy solution adopted is to offer the user the option of asking for suggestions for the whole word or for parts of the word. If *hurli-burli* was queried, you would ask for suggestions for the whole thing; for *computer-desined,* you would ask for suggestions only for the second part.

Quite a large proportion of the program module that deals with input/output is devoted to handling these seemingly minor features of text. It is perhaps a reflection of the trouble that many people have with them in their writing. The former Chief Editor of the Oxford English Dictionaries thinks it's time the apostrophe was abandoned (Burchfield 1985), and the former editor of the Collins English Dictionary would like to take a hatchet to the hyphen (Hanks 1988). These items of quasi-punctuation seem to be less firmly entrenched than the quirks of the orthography – the use of capitals, for instance, has changed considerably since the eighteenth century and the use of hyphens today is variable – but, for the moment, a text-processing piece of software will have to cope with them.

Notes

1. The Lancaster project which produced the tagged, million-word corpus is described in Garside et al. (1987). This work was carried out in association with the universities of Oslo and Bergen, and the corpus is known as the LOB (Lancaster-Oslo-Bergen) corpus. The tagged corpus is described in detail by Johansson (1986). The tagging was carried out mostly by computer and used a table of tag-pair frequencies, like the one described above, derived from an earlier tagged corpus produced at Brown University in America and known as the Brown corpus (Francis and Kučera 1982).

2. Atwell and Elliott (1987) describe various ways of using tag analysis to detect real-word errors, including one that depends, like mine, on detecting improbable tag sequences, though they don't actually include the simple method that I have implemented.

3. The word-frequency lists I used to classify words as common were those from the LOB corpus (Hofland and Johansson 1982), the Brown corpus (Kučera and Francis 1967), Thorndike and Lorge (1944) and Carroll et al. (1971).

4. This was true in the late 1980s, but there has been much corpus-building activity since then, and the large corpora of the 1990s are of the order of a hundred million words. I don't know if corpora of that size are large enough to provide reliable figures on word rarity. The corrector's three categories of word frequency – very common, ordinary and rare – are crude and I think it would be useful to use a more fine-grained system. The corrector has difficulty putting candidates in the right order if they are similar in sound and

appearance, such as *council* and *counsel* (noun). Neither of these words is rare, but *council* must be more common than *counsel* and it would help if the corrector had this information. I hope to include it in a future version.

5. Actually the data structure functions as a queue as well as a binary search tree. I do not allow the tree to grow indefinitely so, at some point, I begin removing old words to make room for new ones.

6. The structure described here bears some resemblance to an augmented transition network (Woods 1970), in that there are conditions and actions on some of the arcs, but there is no recursion in the syllable counter and the conditions all refer to other parts of the input string – they are equivalent to extra arcs.

7. The collections I used were the one described in Chapter Four, plus those described by Hotopf (1980) and by Wing and Baddeley (1980).

8. The problems of dealing with full-stops after initials and so forth are discussed by Booth (1987) in Garside et al. (1987).

CHAPTER ELEVEN

A comparative test and possible developments

The day of reckoning. Does the prototype actually perform any better than the best of the commercial spellcheckers currently available? To find out, I ran some spellcheckers over two test files, and I present the results below.

The first test file consists of short passages taken from *English for the Rejected* (Holbrook 1964), a book about the value of creative writing with secondary school children of low academic ability. The book contains a number of extracts from the written work of some fourteen-year-olds in a secondary modern school in the 1960s, with the children's original spelling and punctuation. The test files are given in full in Appendix Three. Here are the first few lines of the first:

> One Saturday I though I would go to the Races at London I went on my Royl Enfield they can go quite farst. there were the new Japannese Hondor they are very farst and geting quite popular in England I saw one man come off he was on a B.S.A. One man had to go in Hospital because he broke is lege a nouther man hearte is arm in a sidecar race I dont thing it was much I sow a side car turne right over the side care man was alrigh. but the motor bike Rider was thrown into the crowd I dont no what hapend to him he was heart quite bad.

The second test file contains a number of misspellings taken from office documents, all of which were produced by the same person. The misspellings are placed in sentences to give them some plausible immediate context, but the file as a whole is not meant to make sense. Here are the first few lines of the second file:

> When doing the eveluation, I found that it is not fulley compatable with the curent release. Unforchanley this was not known. They have sent us a pacth

> but I am still disiponited with the quility. Support was given on ths instlation and testing. There are no major isues. The second releace has been recived. Tests are beening carred out. It has been checked for dammage and passes the sceond level check sucsesfuley.

These files represent the sort of material that I would like my spellchecker to cope with, but these passages were *not* among those that I analysed in the course of producing the spellchecker; that is to say, I was not testing the prototype on material already familiar to it.

Inspection of the full test passages will show that it is not always easy to decide whether an error should be counted as a misspelling or as an error of some other kind. While *Royl* and *farst* are clearly misspellings, some people might regard the *climb* in 'Bob did not know what to do he climb in the cab' as a grammatical slip, though omitting an inflection is a typical slip of the pen, while others might consider *mums* in 'my mums got a bludy good Nos' as an error of punctuation rather than spelling. I did not count words whose only error involved capitalization of the initial or omission of the possessive apostrophe. I did not count missing words and I did not count a few occurrences of the wrong tense or number that seemed possibly to be non-standard uses rather than mistakes. Otherwise, I counted just about every wrong word as a misspelling – presumably this is what a user would want a spellchecker to do.

There were seven spellcheckers in the test:

Grammatik IV 1.0	on an IBM PC compatible
Wordperfect 5.1	on an IBM PC compatible
Microsoft Word 4.0	on an Apple Macintosh
Wordstar 6.0	on an IBM PC compatible
Macwrite 2	on an Apple Macintosh
Franklin Language Master	on an IBM PC compatible
the Prototype	on a VAX 6310

I used the versions of the software that were current at the time I did the test – the latter part of 1991.[1]

The prototype is a piece of research software running on a multi-user minicomputer, while the others are all commercial products running on single-user PCs. The prototype is just a vehicle for testing methods of correction; it does not pretend to compete with the others in terms of presentation of output,

usability and additional functionality.

I ran each spellchecker over the files, noting the errors that it spotted and any false alarms that it raised. For each error spotted, I noted where the required word came in the list of suggestions (if at all). I attached some significance to the ordering of the lists because I had in mind the needs of a poor speller. I doubt if poor spellers find it helpful to have the required word buried somewhere in a list of, say, twenty suggestions; they would like it at the top. The ordering of the lists, however, was not necessarily something to which the commercial spellcheckers themselves gave much attention. Grammatik, in particular, gave none at all since its suggestions came out in order of length (shortest first) and alphabetically within length. The results of the test are presented in Table 11.1.

There are always two ways of evaluating how well a job has been done – either in terms of what needed to be done or by comparison with other attempts at doing the same job. By comparison with the others, the prototype performed well; it spotted slightly more errors and made a better job of correcting them than any of the other spellcheckers. In terms of the job to be done, however, all the spellcheckers, including the prototype, fell a long way short of perfection; in the young people's writing in particular many of the errors were missed and many of those that were spotted were not corrected.

While perfection is the goal to be aimed at, it would be a mistake to imagine that human beings perform this task with one hundred per cent accuracy. The first of the test passages (the young people's writing) was given to a group of students studying for the GCSE English at a College of Further Education. The best of them detected and corrected about ninety per cent of the errors, the worst about sixty. Taking all the nine students together, they detected and corrected seventy-three per cent of the errors. They missed twenty-four per cent; for the remaining three per cent they offered either another misspelling in place of the error or no correction at all. They also occasionally took a correct word and 'corrected' it to a misspelling, such as *motor* to *moter* or *rosy* to *rosey.*

As compared with the students' score (about three-quarters of the errors detected and corrected), the corresponding figure for even the best of the computer spellcheckers was about two-fifths.

Table 11.1 Test of seven spellcheckers on two files of errors

Young people's writing *(n of errors = 144)*

	Gr	Wp	Ms	Ws	Mc	Fr	Pr
Required word							
1st in list (%)	11	17	23	26	26	26	40
2nd (%)	12	10	10	8	7	13	10
3rd (%)	4	4	4	6	6	3	3
4th – 6th (%)	10	9	2	2	4	4	1
7th – 12th (%)	3	5	0	5	4	2	4
> 12th (%)	0	2	–	2	1	–	–
Not in list (%)	23	13	22	13	13	13	6
Err not spotted (%)	37	40	39	38	39	39	36
False alarms (n)	10	2	6	4	4	4	2

Office documents *(n of errors = 185)*

	Gr	Wp	Ms	Ws	Mc	Fr	Pr
Required word							
1st in list (%)	20	40	49	49	50	53	66
2nd (%)	16	9	6	8	8	10	6
3rd (%)	6	6	2	4	4	4	4
4th – 6th (%)	8	7	1	2	2	3	6
7th – 12th (%)	4	2	1	1	2	4	2
> 12th (%)	1	0	–	0	0	–	–
Not in list (%)	33	18	23	19	18	9	4
Err not spotted (%)	12	18	18	17	16	17	12
False alarms (n)	6	2	3	3	3	2	3

Gr = Grammatik
Wp = Wordperfect
Ms = Microsoft Word
Ws = Wordstar
Mc = Macwrite
Fr = Franklin
Pr = Prototype

Notes on Table 11.1:

1. All the figures except those for false alarms are percentages. Taking the Grammatik column as an example, out of the 144 errors in the young people's writing, Grammatik spotted the error and offered the required word first in its list for 11%. For 12% of the errors, the required word was second in its list, for 4% it was third and so on. For 23% it spotted the error but did not offer the required word in its list of suggestions, while for 37% it did not spot the error at all. The numbers of False alarms are actual numbers of errors. For example, 10 words in the young people's writing that were in fact correct were queried by Grammatik.

2. Although it makes only a small difference to the figures, it is types rather than tokens that are counted in the table; *lory,* for example, is counted once though it appears four times in the test file. With the exception of Grammatik and the prototype, the spellcheckers took no account of context.

3. Wordperfect frequently offered more than twenty suggestions (for *cort* it offered seventy-one, *caught* not among them – the full list is given in Chapter Seven). Microsoft Word offered up to nine. Grammatik, Wordstar and Macwrite usually offered fewer than ten but sometimes more than twenty. The Franklin Language Master and the prototype offered up to twelve.

4. All the checkers incorrectly queried *B.S.A., Gascoyne's, PCs* and *LAN.* All but Wordperfect queried *scrumping* and all but the prototype queried *Enfield. Melton, Clare, DFT* and *PQR* were not counted as false alarms since they were all invented names in the context of these texts, though *Clare* happens to be a reasonably common name and was accepted by both Wordperfect and the prototype. Microsoft Word also queried *James, Tom, Fred* and *modems* while others queried *sixpence* or *program.*

5. The Franklin Language Master is a corrector but not a checker, i.e. it does not check text for misspellings but will offer a set of suggestions for a misspelling that is presented to it. I used it in conjunction with the Wordstar checker; Wordstar found the errors and the Franklin Language Master offered corrections for them. On three occasions, Wordstar found what it considered to be an error but the Franklin offered no corrections since the word was in its dictionary – the second *to* of *to to, Nos* (plural of *No*) and *program.*

6. Grammatik checks grammar and style as well as spelling. Its grammar checking part was occasionally misled by the poor punctuation of the young people's writing into querying some correct words, though it might be argued that these were not really false alarms since it was indirectly drawing attention to the errors of punctuation. Since the grammar checker could be expected to work better on a text free of spelling errors, I made a second pass through the files, after correcting all the errors to which it had drawn attention on the first pass. It found three more real-word errors but raised two more false alarms.

It should be said on the computers' behalf that the errors in this passage were mostly of the kind that people find easy and computers find hard: errors in fairly common short words or where the context is important – *thort, lsning, hearte* (for *hurt*), *spund* (for *spun*). If the passages had contained more minor errors in long words (like *indiscrest*), the students would have done worse and the computers better. It can also be said that the admittedly rare human error of offering a misspelling (i.e. a non-word) as a correction is one that the computer spellcheckers never made.[2]

On the side of the students, I should add that many of them carried out a more general editing job than the computer spellcheckers did, attending to punctuation and even to sentence structure as well as to spelling. There is no doubt that, on these particular passages, the human spellcheckers performed a better job than the computer ones.

It is clear that there is much room for improvement. There are a number of ways in which the performance of the computer spellcheckers, more specifically the prototype, might be improved, and I will round off the chapter with a number of suggestions. Some of them are extensions of techniques used in the current version, requiring only time and effort to incorporate; others are more speculative.

The dictionary

A spellchecker is only as good as its dictionary. The prototype's dictionary is based on one derived from the Oxford Advanced Learner's Dictionary of Current English (OALDCE). The Oxford University Press made available the machine-readable text of this dictionary for research work via the Oxford Text Archive, and I derived a computer-usable version from it containing spelling, pronunciation and word-tags (Mitton 1986). A test of this version of the dictionary was conducted at Leeds University (Sampson 1989) in which a sample of fifty thousand words was taken from a million-word corpus of general text (the LOB corpus – see Chapter Ten) and a computer looked up every word of the sample in the dictionary. A little over three per cent of the words were not in the

dictionary. However, many of these were items that you would not expect to find in a dictionary, such as obscure names, foreign words, mathematical formulas and other odd things, and some others were hyphenated words where the separate parts were in the dictionary (*blue-painted, coffee-blending*) but not the hyphenated item as a whole. The unhyphenated English words that were not in the dictionary accounted for well under one per cent of the sample, and, even among these, some were decidedly obscure while others were hyphenation problems in that the dictionary contained them in hyphenated form (*co-operate, cubby-hole*) whereas they appeared unhyphenated (*cooperate, cubby hole*) in the sample.

These results were reassuring, but they also pointed to some weaknesses in the dictionary that needed attention. One was a shortage of *un* words, such as *uneven, unlocking, unrelated, uncomfortably.* The OALDCE was intended, of course, for people rather than computers, and a human reader could presumably look up *comfortably* and conclude that *uncomfortably* meant the opposite, so this was not a serious deficiency in the original dictionary, though for some of the missing *un* words (such as *unassailable* and *uncomprehending*), the positive form is rare or nonexistent. Other weaknesses shown up by the Leeds University test include a shortage of *-ly* words (*inevitably, ritualistically*), *-ness* words (*fairness, wholeness*) and *re-* words (*reassembled, resell*).[3]

It is partly because of dictionary deficiencies of this kind that some checkers make use of affix-stripping, described briefly in Chapter Seven. For example, even if *shortness* were not in the dictionary, an affix-stripping checker would still accept it because, after removing the *ness,* it would find *short* in the dictionary. I had to make a decision on this when building the dictionary, since an affix-stripping checker needs only word-stems whereas a full-word checker needs all the derived and inflected forms entered in full.

I decided against affix-stripping for two reasons. One was that a checker that accepted *shortness* would also accept *longness,* which perhaps it ought not to do; the problem of real-word errors, already serious, would be aggravated by quasi-real-word errors. The other reason was that the spellchecker was going to use the dictionary for producing suggestions as well as for checking. If *shortness* were not in the dictionary, how could it produce *shortness* as a candidate for, say, *shortnace*?

Another possibility would be to strip an apparent affix from a misspelling, to produce candidates for the stem, and then to stick the affix onto the candidates. For example, given *dedness,* the spellchecker could find candidates for *ded*, such as *dead* and *deaf*, and stick *ness* on them to produce *deadness, deafness* and so on. But this runs a serious risk of producing words that are very rare or nonexistent – *dreadness* would appear in the above list – even assuming that the affix-sticking took account of part-of-speech and therefore was prevented from producing *debness, denness* and the like. Given that there were no tight limits on the dictionary's storage space, it seemed safer to deal with derived forms and inflections when building the dictionary rather than risk the pitfalls of affix-handling at run-time.

Another useful point to emerge from the Leeds test was that the dictionary's stock of abbreviations could be much enlarged. (I have now done this.) Beyond that, there is no simple way of getting a list of words to add. One way would be to inspect all the words that occur in a corpus but are not found in the dictionary and to pick out those that ought to be included. I have done this with words from the LOB corpus and added about twelve hundred words as a result. Almost all of these were derived forms of words already there (which makes me wonder if I made the wrong decision about affix-stripping). Another way would be to compare the dictionary against other machine-readable dictionaries to see whether there are any words in the other dictionaries that are not in the OALDCE and ought to be. Like all sizable improvements to the dictionary, this work would not be difficult in principle, just tedious. It could and should be done, but the tests against the LOB corpus showed that the coverage of the present dictionary is pretty good; enlarging the dictionary would make only a small improvement to the spellchecker's performance.

Improving the checking

The results of the comparative tests showed that it was mainly in the detection task that the computer spellcheckers fell down. The errors they failed to spot were all real-word errors. Except for

Grammatik and the prototype, they would detect a real-word error only if their dictionary did not happen to contain the word. For example, most of them had the word *toner* in their dictionaries and therefore failed to query it when it appeared in context as a misspelling of *tonner* (a sort of van, a *ten tonner*). Microsoft Word did not happen to have this word in its dictionary and so correctly queried it. This total reliance on dictionary look-up, along with the large size of their dictionaries, meant that they detected hardly any real-word errors. (In fact the presence of some highly obscure words in their dictionaries meant that they failed to spot some straightforward errors; Wordperfect, for instance, failed to query *lege, sone,* or *alow*.)

Grammatik's grammar checking enabled it to spot a few real-word errors, though at the cost of raising a number of false alarms. The prototype did a little better than the others because it performed at least a small amount of syntactic checking, based on the table of tag-pair occurrences derived originally from the LOB corpus. It spotted 'I have spoken to customer support on this *be* they could ..', 'a problem with the program *were my is* can ..', 'I have noticed *the* there is ..', 'ask people for *there* name ..', 'There *seams* to be ..', 'could you *advice* me ..', and 'should also be *enable.'* It also spotted *Some one* and *With out* because of its check for commonly split words. (It also spotted *toner* and *corer* but this was just because these words were not in its dictionary.) But this leaves about seventy real-word errors that it missed. I have not done extensive tests, but my impression is that these figures are typical; in other words, it detects around ten per cent of real-word errors on the basis of syntactic context. Various changes could be made to improve this rate, both to the table and to the way the program uses it.

As I described in Chapter Ten, I have already made small changes to the table to improve its error-spotting by inspecting all tag-pairs that occurred just once in the LOB corpus and changing the corresponding table entry, in some cases, to zero because the occurrence was the result of a mistake or a very peculiar use of English. I think this process could be taken further, looking at pairs that occurred, say, two to five times. For example, there were a handful of occurrences of nominal adverb (*here, now, there, then* etc) followed by noun, so this pair was recorded as rare-but-acceptable

in the table. This prevented the checker from querying *there name* and the like, so I changed the table entry to zero ('unacceptable') in order to catch this frequent error. It might be worth risking a few false alarms by doing this to a few more pairs.

A more ambitious alteration to the tag-pair table would be to modify the set of tags. There is information in the computer-usable dictionary – the one from which the spellchecker's dictionary is derived – which is not presently being used. In particular, nouns are tagged as countable or uncountable (or both) and verbs are tagged as transitive or intransitive (or both). The nouns and verbs of the LOB corpus are not divided up in this way. It would be possible to go through the LOB corpus, looking up every noun in the dictionary, classifying it by countability, and deriving a different set of tag-pair frequencies for nouns (countable), nouns (uncountable) and nouns (both), and similarly for verbs with respect to transitivity. It is possible that the frequency distributions for countable and uncountable nouns, and for transitive and intransitive verbs, would be sufficiently different to provide the spellchecker with useful extra information for error-spotting and candidate-ordering.

Another, less ambitious, modification along the same lines would be to take a few very common words, such as *of* and *in*, out of the classes to which they are presently assigned and give them tags of their own. It may be that the distribution of these words differs in some useful way from that of their current tags.

One reason why the syntactic error-checker detects few errors is that it has a strong bias towards caution. It will flag an error in the string of words from A to B only if there is not a single acceptable sequence of tags linking the two; in other words, it will flag an error only if there is *no* interpretation of the words from A to B that makes syntactic sense. As described in Chapter Ten, the tag pairs are rated in its table as Acceptable, Rare, Very rare or Never. The checker could investigate those strings of words where the only tag pair sequence that was acceptable contained a pair rated Very rare. If it found, furthermore, that one of the words in this pair appeared only rarely with this tag, it could perhaps query the pair of words without running too great a risk of raising a false alarm.

Improving the correcting

When human beings are correcting a passage, they make great use of the sense of the words; *currant* might be corrected to *current* in a piece about electricity, but vice-versa in a recipe. Computer spellcheckers do not have this kind of understanding at all. It might be worthwhile incorporating some semantic, or at least collocational, information in the dictionary, to help the spellchecker to order its suggestions.

The creation in recent years of very large computer-readable corpora, of the order of a hundred million words, has enabled researchers to try a head-on approach to this problem. In Chapter Seven I described some work by researchers at IBM using word triples (Mays et al. 1991). Given any two words they can give an estimate of the probability of any particular word occurring next; given *I think,* for example, they could say what the probability was of getting *that* as the next word. I described an experiment in which this table was used both to detect and to correct real-word errors. Similar work at Bell Labs has produced a table of probabilities for word pairs; given *I,* they could say what the probability was of getting *think* as the next word (Gale and Church 1990, Church and Gale 1991). This table is used to reorder the list of candidates produced by a spelling corrector. For example, for the misspelling in 'a versatile *acress* whose combination of ..', the corrector produces *acres, actress, across, access, caress, cress.* Consulting the word-pair table enables the system to put *actress* at the top of the list since the corpus contained two occurrences of *versatile actress* and eight of *actress whose* but none of *versatile acres* or *acres whose.*

There are problems with these methods, mainly in the treatment of words or pairs or triples that never occurred in the corpus, of which there are many. Assigning them a zero probability seems unsatisfactory – who is to say that a valid occurrence won't appear in the next addition to the corpus? Besides, it gives poor results. But assigning them an appropriate non-zero probability requires some care; the Bell Labs work shows that poor estimates of these probabilities are worse than none, i.e. they can actually lead to a worse ordering of the candidates.

Though these methods provide enormous detail about the likely neighbours of individual words, they are restricted to near neighbours. Useful semantic context might be further away. What if the clue to *current* were the word *voltage* in the previous sentence?

One way to take advantage of such clues would be for each word in the dictionary to have a list of words that were likely to co-occur with it – 'collocations', to use the linguists' term. These might be neighbouring words in set phrases, such as *fish* with *chips* or *package* with *holiday,* or words more loosely associated; the word *gap,* for example, tends to occur close to *teeth, mountain, record, years, poor, rich, trade, generation, narrow, widen, fill, close* and *reduce* (Moon 1987). Given a list of suggestions for a misspelling, the corrector would search the text for some distance (say ten words) to right and left of the misspelling to see if any of the candidates' associated words occurred. If they did, the corrector would move the candidates up or down the list accordingly.

The main problem with this scheme is getting the lists of associated words into the dictionary. One way would be to have a computer go through a machine-readable dictionary, such as the OALDCE, deriving lists of words from the definitions (and perhaps from the examples also). Of course, a definition is not the same as a list of collocations; the OALDCE definition of *boast,* for instance, is 'words used in praise of oneself, one's acts, belongings etc,' which would probably be of little use for the purpose I am considering. But some other definitions are more promising; the definition of *boat* contains *travel, water, oars, sails, fishing, crew, passengers, sinking, ferry* and *pleasure,* any of which are quite likely to co-occur with *boat.* A small piece of work on disambiguation (i.e. not spelling correction but getting a computer to decide which sense of a word was being used) used dictionary definitions very much in the way I have described and found them surprisingly useful (Lesk 1987).

The best way to produce the lists of associated words, however, would be to take all the occurrences of a word, with context, from a very large corpus, such as the one collected by the COBUILD project (Sinclair 1987) or the British National Corpus recently assembled by a consortium led by the Oxford University Press (Leech 1993) and to derive a collocation list directly from the contexts.

This last suggestion – the incorporation into the dictionary of lists of collocations – is a good example of a general problem that now faces automatic spelling correction. Whichever method was used, generating the collocation lists would require some care, and the resulting lists would increase the storage space occupied by the dictionary, even assuming that the lists were stored in some compressed form and not literally as lists of fully spelt-out words. By contrast, the effect that it would have on the spellchecker's performance, though presumably in the desired direction, would be fairly small.[4] Table 11.1 shows that when the prototype produces the correct word somewhere in its list of suggestions, it generally offers it high up the list; there is not all that much room for improvement here.

Spelling correction is subject to the law of diminishing returns. A quite simple system will detect about two-thirds of misspellings and may offer corrections for many of them if they are predominantly typing slips. Building in some knowledge of pronunciation produces a significant improvement in the spellchecker's ability to come up with the desired word, especially for the misspellings of poor spellers. After that, it's all uphill. Large increases in the program's knowledge and sophistication produce relatively small improvements in performance. It is encouraging, however, that each additional item of information – pronunciation, tag-pair probability, word frequency, collocations and so on – seems to improve performance, albeit by only a little, rather than degrade it. The spellchecker that is as good as a good typist is not yet a reality, but there is no reason to think that it is only a dream.

Notes

1. More recently (March 1995) I repeated the test with two of the more widely used spellcheckers – Wordperfect 5.1 and Microsoft Word 6.0, both running on a 486DX2 PC. The results for Wordperfect were exactly the same as in 1991; those for Microsoft Word had improved, as summarized in the following table:

	Children	*Office*
Required word		
1st in list	30%	59%
2nd	6%	9%
3rd	2%	2%
4th–6th	3%	1%
7th–12th	2%	1%
>12th	0%	0%
Not in list	17%	11%
Error not spotted	40%	17%
All errors (=100%)	144	185
False alarms	4	0

2. Though this human error may be rare, it can be serious. A doctor friend of mine was alarmed to notice that his temp secretary had replaced every occurrence of *hyper* with *hypo* when typing some reports he had written, under the mistaken impression that she was correcting his bad spelling.
3. These results have been confirmed by one of my MSc students at Birkbeck College who carried out the same exercise using the whole of the corpus as opposed to just a sample (Dougal 1991).
4. A small experiment carried out by another of my MSc students suggests that the use of collocation lists would improve the spellchecker's ability to get the right candidate at the top of the list but that the improvement would be small (Vaidya 1992).

APPENDIX ONE

The prototype implementation

Chapters Eight to Ten described the general ideas incorporated into the corrector. This appendix gives some details of their specific implementation in the prototype. Here again is the simplified overview of the program presented in Chapter Eight:

Taking the text sentence by sentence:
    Split the input into words and store each word in memory.
    Look up each word in the dictionary, and mark it if not found.
    Check each pair of words for anomalies of syntax.
    Display sentence, possibly with queries.
    If any words have been queried, then
        For each queried word, do:
            Generate list of suggestions.
            Offer best few to user.
            Get user's decision.
            Insert user's choice in sentence.

The procedure described as 'Generate list of suggestions' has three main parts:

1. Retrieve misspelling's own S-code group and string-match all the words from length x to length y against the misspelling, putting the best into an ordered shortlist.
2. Retrieve related S-code groups; for each of these, take the words from length x to y and do a quick test on their letter-strings, passing on the successful few for string-matching and possible addition to the shortlist.
3. Reorder the shortlist by word-tags and word-frequency.

The prototype was written in Cobol. This is a language more associated with file processing in business and administration than with natural-language applications, but it was obvious from the beginning that the corrector would require random file access, and Cobol provides this as a feature of the standard language. It is provided in many other languages – Pascal, for instance – only as a manufacturer's extension to the standard, and I preferred to work in a standard language, rather than in some dialect specific to one manufacturer. The 1985 Cobol standard meets many of the criticisms levelled at earlier versions of the language, and it even includes some simple features for string-manipulation. I also considered Icon (Griswold and Griswold 1983), a language with powerful string-handling facilities, but decided against it, partly because of its weakness in file-access, but also because, paradoxically, I did not have much use for its string-handling features. The string-matching that produces the closeness-score was something I had to program myself in detail; the program's other string-handling is fairly straightforward.

An early version of the prototype, built along the lines described in Chapters Eight to Ten, made a reasonable job of detecting and correcting misspellings, but it was rather slow. It was never intended that the prototype should perform as fast as a commercial piece of software. Nonetheless, I did not want it to be purely an academic exercise; I wanted to show that it had at least the potential to be turned into a real-life spellchecker. The rest of this appendix describes some modifications that were introduced to speed it up.

An elementary observation to be made about ordinary prose is that a small number of words occur a great many times. The prototype holds the 1024 words that occur most frequently in running text in a Cobol table in main store. When looking words up in the dictionary, simply to establish whether they are there or not, the spellchecker first consults this table, which is searched by binary search (using the Cobol SEARCH verb), and only proceeds to consult the dictionary file on disc if it fails to find the word in the table. (Retrieving a record from secondary storage takes a lot longer, of course, than doing a binary search on a table in main store.) Taking the first sentence of this paragraph as an example, the spellchecker would need to consult the disc file only for

elementary, *observation*, *prose* and *occur*.

In the first version of the dictionary file, each word occupied one record; if the spellchecker needed to look at a hundred words, it retrieved a hundred records. Using the simple timing facilities offered by VAX/VMS (the system under which the corrector was developed), I discovered that the program spent over half of its time – and this was CPU time, not elapsed time – carrying out these READ operations. Some simple experiments showed that it was far quicker to pull in a hundred words as a single large record than as a hundred small ones, so I reorganized the dictionary file to take advantage of this.

When the corrector searches the dictionary for promising candidates, it considers sets of words that are in the same Soundex-type group, so the obvious thing was to turn each of these groups into a single, variable-length record. This provided the opportunity to give each of these large records some internal structure, as follows:

> Soundex-type code (the record key)
> Fifteen fixed-length pointer-items
> Variable number (50 to 300+) of letter-string-items
> Variable number (50 to 300+) of spelling-fields

The first of the pointer-items refers to words of length one, the second to words of length two, and so on; the fifteenth refers to words of length fifteen or more. Each pointer-item contains this information:

> Number of words of length n
> Position in letter-string section of first word of length n
> Position in spelling-field section of first word of length n

The positions are held as byte offsets from the beginning of the record. The letter-string items are variable length and are separated by low-value bytes; likewise the spelling-fields.

I explained in Chapter Nine how the spellchecker retrieves groups that have codes related to that of the misspelling (so as to succeed with things like *atlogether* and *unerstand*) and how it discards most of the words in them with a quick test based only on each word's letter-string – the letters of the word in alphabetical order, without repeats. For the great majority of words considered

as candidates for a given misspelling, it would be a waste of time to unstring the spelling, word-tags and other fields only to discard them after a quick look at the letter-string, so the letter-strings are held in a separate section of the record. Each letter-string item contains the required information for one word in the following form:

Letter-string
Byte-offset of this word's spelling-field

The final section of the record is a long string containing all the remaining information about the words, with each field divided from the next by a separator. The information for one word is as follows:

A single-digit value called 'start-point' (explained below)
Spelling in coded form (for string-matching)
Letters that might be inserted in this word
Spelling in ordinary form
Word-tag(s)
Number of syllables
Homophone code

Having the letter-string information in a separate section enables the corrector to run through the letter-strings (using the Cobol UNSTRING verb) and to perform the quick test on one word after another as fast as possible. It picks out the other information about a word only for the few words that pass the quick test. By contrast, when considering words in the same group as the misspelling, it performs the string-matching on all the words from length x to length y, so it simply moves along a section of the long string of spelling-fields, unstringing one after another.

These arrangements have the unfortunate effect of making it more difficult to simply look up a word. The spellchecker computes the word's Soundex-type code and retrieves the appropriate record. Then it computes the word's letter-string, takes that part of the letter-string section that contains words of the right length and searches it for a match. If it finds one, it unstrings the corresponding spelling and compares it with the word being looked up. If they don't match, it carries on. The letter-string items are arranged in alphabetical order so that the corrector can

abandon the search if it finds it has gone past the place where the word's letter-string would have been. Despite these convolutions, the dictionary look-up still takes very little time compared with the retrieval and ordering of candidates for a misspelling.

This reorganization of the file reduced dramatically the time required for the mere retrieval of records from the dictionary; when run over a file of test data, the program took only about forty per cent of the time that it had taken with the previous file organization. The part that now took up most of the corrector's time was the string-matching that produced the closeness scores.

To recap briefly on Chapter Eight, the string-matching is regarded as the traversing of a directed network. Computationally, each node of the network is represented as an element in a two-dimensional array, and the algorithm takes account of all paths across the network by computing values for the elements of the array in an ordered sequence. The computation of the value for a single element in the array requires the calculation of four values (corresponding to a single-letter omission, insertion, substitution or transposition), and each of these entails taking a number from a table and adding it to one of the array values already calculated; the lowest of the four is retained as the value for the element.

The reason why this takes time is not that the calculation of an array value is slow – Cobol indexed tables are used and all variables involved are specified as COMPUTATIONAL – but that there are so many array values to be calculated. The size of the array depends on the length of the misspelling and the length of the candidate being compared with it; a misspelling and a candidate that were both of length eight would require an array of eighty-one elements (nine by nine). The matching of this misspelling against a hundred candidates – some shorter, some longer, some the same length – would require the computation of about eight thousand array values. Attempts to reduce the time spent on string-matching therefore focus on ways to cut down the number of array values to be computed.

The mincost function presented in Chapter Eight computes the array values row by row. Take the first three columns of two rows to be represented by the letters *u* to *z*, as follows:

u v w
x y z

Ignoring transpositions for the moment, the value z is the lowest of the following:

w + an omission cost
v + a substitution cost
y + an insertion cost

Costs are either zero (for substituting, say, *a* for *a*), or positive. So:

z >= min(v,w,y)

The value y is in the same position with respect to u, v and x:

y >= min (u,v,x)

And x is u plus an omission cost (positive), so:

x > u

It follows that:

y >= min(u,v)
z >= min(v,w,min(u,v))
z >= min(u,v,w)

And, in general, the lowest value on row i cannot be lower than the lowest value on row i–1. In other words, a candidate's score cannot get better with each row calculated; it can only stay the same or get worse.

Transposition costs could spoil this picture. Transposition provides a further way of calculating the value z as follows:

r	s	t
u	v	w
x	y	z

r + a transposition cost

Suppose that r had the value 3, that 7 was the lowest value in the u-v-w row and that the transposition cost was 3. Then z could have the value 6, lower than the previous row's lowest.

This can be prevented by setting the transposition cost to be at least as large as the largest omission cost. Suppose that both were set at 5. This clearly avoids the problem in the example since the transposition route to z now costs 8. In fact it avoids it altogether since it is always possible to take the omission route from one row to the next, which will never cost more than 5, so the transposition route from row i–2 to i cannot now cost less than the lowest-cost route from row i–2 to i–1.

The importance of all this is that it enables the corrector often to abandon the calculation of array values after doing only the first few rows of an array – a simple application of the 'branch and bound' technique. When considering a hundred candidates, it begins by putting the first fifteen into its shortlist. Thereafter, it puts a candidate into the list only if the closeness-score is better (lower) than that of the worst candidate in the list so far. Suppose it were considering the seventieth candidate, and the worst candidate in the shortlist had a score of twelve. Suppose also that the lowest value in row five for this (the seventieth) candidate was fourteen. Then there would be no point in continuing the array calculations to the end since the final score could not be less than fourteen. This candidate is obviously not going to make the shortlist, so it can be rejected without more ado.

A variation on this idea is presented in a recent paper (Du and Chang 1992). Instead of calculating the array values row by row, the authors suggest calculating them 'layer by layer'. What they mean by this is illustrated below.

1	2	3	4
2	2	3	4
3	3	3	4
4	4	4	4

Beginning at the top left, you calculate value 1, then the values marked 2, then those marked 3 and so on. What I have just described for rows also applies to these layers. A candidate's score in layer 4 cannot be lower than its lowest score in layer 3.

The next time-saving modification builds on the observation that words in the dictionary often come in sequences that begin with the same first few letters. This does not happen as much as it

would in a dictionary that was completely in alphabetical order, but, even in the prototype's dictionary, there are short runs of words in the same Soundex group and of the same length that begin with the same letters. Given the misspelling *undeterd*, for example, all the words in group U533 would be considered from length 6 to length 10. Suppose that *undertake* was followed by *undertook*, both being matched, of course, against the same misspelling. There would be no need to compute the first few rows for *undertook* (the ones corresponding to *undert*) since they will be exactly the same as they were for *undertake*. To facilitate this, each dictionary entry carries a number – the 'start-point' mentioned above – telling the corrector how many rows of the array it can skip, assuming that the previous word in the dictionary was the last word it dealt with.

The corrector is here saving a little time by making use of the state of the array left over from the previous call to the string-matching function. This is no problem in Cobol since local variables retain their value from one call to the next.

The size of the array increases as the square of the length of the misspelling, so it is particularly important to reduce, if possible, the number of array values to be calculated in the larger arrays. In addition to slicing off the bottom and the top of the array, as described above, the corrector also cuts off the corners.

In traversing the directed network from the start node to the diagonally-opposite end node, a route that takes in either of the other two corners is most unlikely to be a low-cost route since these routes correspond to lots of insertions followed by lots of omissions (or vice-versa). A low-cost traversal is almost certain to contain a number of zero-cost substitutions, which means it will stay fairly close to the diagonal. For larger arrays, therefore, the corrector computes values only for a diagonal band across the array, of about three elements to either side of the diagonal, and ignores the elements outside this.[1]

Compared with the dramatic difference in speed produced by reorganizing the dictionary file, the effect of these program modifications is rather modest. Each one reduces the running time by about five per cent, though the precise effect depends on the actual misspellings being corrected. To give some idea of the speed of the prototype, I ran it over the following four sentences (with

one misspelling per sentence):

1. It's a *granuler* substance.
2. This is the *sought* of thing we want.
3. You can *chooce* whichever you want.
4. I can't *onderstan* it.

Running on a VAX 11/750 under VMS, the CPU times were as follows:[2]

		1	2	3	4	Total
	Total CPU seconds	1.7	2.6	3.6	5.5	13.4
	of which:					
a)	string-matching	0.5	0.9	2.5	2.1	6.0
b)	quick comparison of letter-strings	0.7	1.1	0.7	2.3	4.8
c)	other	0.5	0.6	0.4	1.1	2.6

Elapsed times, of course, are larger than CPU times, but they are not worth reporting since they depend almost entirely on the amount of work the machine happens to be doing for other users.

The bulk of the time goes into producing the lists of suggestions. The time-consuming parts, as the table shows, are the string-matching and the comparison of letter-strings, but some of the time marked 'other' also goes into producing the lists of suggestions – for example reading records from the dictionary file and reordering the shortlist by word-tag and word-frequency. The variation from one misspelling to another is explained by the number of candidate words in the misspelling's Soundex-type group, the number of neighbouring groups that have to be searched and the number of words in these neighbouring groups. *Granuler* belongs to the smallest group (G540), whereas *chooce* and *onderstan* belong to two of the largest.

Notes

1. Herewith a cautionary tale for program optimizers. Since the array is not always square – you might be comparing a twelve-letter candidate with a nine-letter misspelling – my first attempt at this included some calculations to

establish which elements fell on the diagonal. This version actually ran slower than the one it was supposed to be improving on. The problem was that the calculations included a division, which is a computationally lengthy operation. A less elegant but simpler version had the desired effect.

2. I include these times to give some idea of the relative lengths of the various operations. The College has now replaced the VAX 11/750 with a VAX 4100 and the spellchecker runs much faster.

APPENDIX TWO

A list of function words

The distinction between function words and content words is one often made by psychologists. For most words, the distinction is easy to make – *of* is obviously a function word and *antelope* obviously isn't – but there are a number of borderline cases. It is not obvious to me whether *everything, notwithstanding* and *underneath,* to give just three examples, are function words or not. A further complication is that a single spelling may represent sometimes a function word ('*Will* you go?') and sometimes a content word ('last *will* and testament'). Other examples include *going,* which looks like a content word in 'She's going too fast,' but a function word in 'She's going to meet a sticky end,' and *used* in 'She used a spanner,' and 'She used to play the harpsichord.'

Not wishing to get involved in these complications, I simply adopted a list of words (i.e. spellings) which Professor Frank Knowles of Aston University kindly supplied to me. It is an enlarged version of one published by Margaret Masterman as an appendix to a paper (Masterman 1979). I have added the contractions of verbs with *not – can't, won't* and so on.

A LIST OF FUNCTION WORDS

I	away	elsewhere	how
a	back	enough	however
aboard	backward	even	if
about	backwards	ever	in
above	be	evermore	indeed
across	because	every	inner
after	been	everybody	inside
again	before	everyone	instead
against	beforehand	everything	into
ago	behind	everywhere	is
ahead	being	except	isn't
all	below	fairly	it
almost	between	farther	its
along	beyond	few	itself
alongside	both	fewer	just
already	but	for	keep
also	by	forever	kept
although	can	forward	later
always	can't	from	least
am	cannot	further	less
amid	could	furthermore	lest
amidst	couldn't	had	like
among	dare	hadn't	likewise
amongst	daren't	half	little
an	despite	hardly	low
and	did	has	lower
another	didn't	hasn't	many
any	directly	have	may
anybody	do	haven't	mayn't
anyone	does	having	me
anything	doesn't	he	might
anywhere	doing	hence	mightn't
apart	don't	her	mine
are	done	here	minus
aren't	down	hers	more
around	during	herself	moreover
as	each	him	most
aside	either	himself	much
at	else	his	must

mustn't
my
myself
near
need
needn't
neither
never
nevertheless
next
no
no-one
nobody
none
nor
not
nothing
notwithstanding
now
nowhere
of
off
often
on
once
one
ones
only
onto
opposite
or
other
others
otherwise
ought
oughtn't
our
ours
ourselves
out
outside
over
own
past
per
perhaps
please
plus
provided
quite
rather
really
round
same
self
selves
several
shall
shan't
she
should
shouldn't
since
so
some
somebody
someday
someone
something
sometimes
somewhat
still
such
than
that
the
their
theirs
them
themselves
then
there
therefore
these
they
thing
things
this
those
though
through
throughout
thus
till
to
together
too
towards
under
underneath
undoing
unless
unlike
until
up
upon
upwards
us
versus
very
via
was
wasn't
way
we
well
were
weren't
what
whatever
when
whence
whenever
where
whereas
whereby
wherein
wherever
whether
which
whichever
while
whilst
whither
who
whoever
whom
whose
why
will
with
within
without
won't
would
wouldn't
yet
you
your
yours
yourself
yourselves

APPENDIX THREE

The test passages

These are the test passages that were used for the comparative tests reported in Chapter Eleven.

Young people's writing

One Saturday I *though* I would go to the Races at London I went on my *Royl* Enfield they can go quite *farst.* there were the new *Japannese Hondor* they are very *farst* and *geting* quite popular in England I saw one man come off he was on a B.S.A. One man had to go in Hospital because he broke *is lege a nouther* man *hearte is* arm in a sidecar race I *dont thing* it was much I *sow* a *side car turne* right over the *side care* man was *alrigh.* but the *motor bike* Rider was thrown into the crowd I *dont no* what *hapend* to him he was *heart* quite bad. the sidecar broke off the *moter bike* and *spund* down the track for 7 yards then hit a man who was on the corner and killed him his hat was blowing down the track the *moter bike caugh* fire and *blow* up the *petal* went all over the track but they *sone* put it out by sand there was some oil got on the track as well it made it very slippery. *They* were quite a few people there because it was a nice dry day The *fianl* race was on they were on the starting line Bang *there* off one BSA is still on the starting line the rest are round the bend I can see one man off he is on a *Trumh* I think a Japanese *Hondor* is in the lead. *his one* by 3 yards and a *Royl* Enfield 2 and a *Hondor* 3.

It comes off of a tree when you pick it is rosy red and before you eat it makes your mouth water it even makes your mouth water when you hear the name gascoyne's scarlet. when you go scrumping and when you get caught it is worth taking the chance of getting some gascoyne's scarlet apples. when you go scrumping you have got to be *carefull* you don't get caught if you do get caught the man will be waiting for you next time you go to get some more gascoyne scarlet apples. when I go scrumping all I go after is apples and plums, pears. when you go pinching *apple* you do not want to take *to* many people or you will not get away so quick when there are a lot of people with you *a specally* little boys they make too much noise. If you pick apples off of the ground you have to be *carful* of the wasps or else they will sting you and the sting comes up to a big bump *where ever* it stings you. when you are by yourself and *some one* comes and you have to climb over the *barb* wire and get caught the man will *probally* catch you but if someone else is with you they can unhook you. some people when *the* catch you they might hit you and *said* he will hit you harder next time he catches you. But some people say they will hit you next time but warn you not to come in the orchard again But the children could not resist getting some more apples to eat. It is best getting apples from an orchard where there are no houses. when I go scrumping I do not take bikes because the man who owns the orchard might come along and take the bikes and if you want your bike you have to go and ask for it and that is how the owner of the orchard knows that you were in the orchard. I never take a dark jacket or a red one because you will be *reconnised* very easy because red shows up very easy. But still it is worth getting some gascoyne's scarlet apples.

Bob *dicided* to *to* it. on Sunday night Bob went to the market hill and *pick* up the *lory.* He *new* how to drive it *becaues* he had *drivern* one *befor.* He was about 5 miles out of Melton and he began to think to himself "I *wund whate* is on this *lory* James *semed every* suspicious. *thort* Bob. I *wund whate* is on it. I will have a look when I get to a *la-by.* he comes to a *la-by* and pulled in and had a look at the *lod* it was *stolon* whiskey Bob did not know what to do he *climb* in the cab, and sat and thought to himself, I *wunder* what to do, *shod* I tell the police or *shod* I keep *quiert.* He *dicider* to tell the police so

he *cared* on to a little village near Clare he saw a police car coming *donw* the road it *stop* the policeman wound down the window "You have not seen a man with a *lod* of whiskey on a *Autin* Ten *Toner hav* you." Bob said, " I am glad you *stop becaues* I am driving the *lory* the *lory* is round the *corer* I *stop* here *becaues* I was going to tell the police if you follow me you can catch the gang" The policeman said, "*Led* the way."

One day Tom an old *tamp* was *warking* up a hill, it was getting dark and it was very cold. It must have been *frizing had* for there was icicles hanging *for guter* of *houes,* the *snowe* on ground was *cresp* as he *wark* through the village. Tom *sorw* a light coming *for* a farmhouse it did not look far. Old, Tom jumped a *dich* and *wark towords* the light he *arived* at farm. He could see a *bran* he went in and in corner there was a *heep* of *strow* in another there was a *heep* of *potates* and he could see a rabbit *loker.* He *wark* over to it there was rabbit in it but he could not see how many there was. He went back to the *storw* lay down and went to sleep. Next morning he was up at 4 am it was dark then he *throught* he had *beter* go before the farmer got but he did not go *emty hang* he took two of the farmers rabbit and *sum potates* in a bag.

Smocking in the *houes*
One night in bed I had a puff,
wil reading literature and,
lsning to hear if the *sairs* would *creek*
so I was not *cort indiscrest*
not a sound I could hear
utill the door was open
as quick a *flass* the fag was in bed
But my *mums* got a *bludy* good *Nos*

The cat working in the mill *spys* a *moues* feeding on corn. The cat *scillfully creps* up behind a sack and all of a *suden* his *musirls* are *tens* he *spring* and a little *squick* from the mouse I *herd* as the cats *clors sunck* deep into the mouse the cat *quilly ete* it and then *cerled* up on a sack to *slip.*

This is the tractor *witch* I drive at weekends. It is a low *gird* tractor it has three *fowd girse* and 1 *reveirse.* It has *for silinders* and a *high*

drollit lift. the *many fold as* burnt out so smoke and flames and sparks fly out of the *botton* of it. if you fill the *radyator* up with water it will last for 10 minutes *becouse* it has a hole in the bottom of it as big round as a sixpence.

Office documents

When doing the *eveluation,* I found that it is not *fulley compatable* with the *curent* release. *Unforchanley* this was not known. They have sent us a *pacth* but I am still *disiponited* with the *quility.* Support was given on *ths instlation* and testing. There are no major *isues.* The second *releace* has been *recived.* Tests are *beening carred* out. It has been checked for *dammage* and passes the *sceond* level check *sucsesfuley.* Fred is away *form* the 5-5-89 and this leaves us *vary* exposed. At our recent meeting you *statied* that a rewrite of the *moninter* package was being *planed.* Here is a list *compilde* after *speeking* to my *colliges.* We want *supprot* for the *statictics,* local *referances* and *circit* numbers and *interreactive* mode while *veiwing* screen. I pass these *commints stright* to you. Given my lack of *knowalge,* it may be *posible* to do these already. Let me know when you can *commonit,* so I can provide *feed back* to the DFT. Can you give me a *formular* to *calulate* the number? I *beleave* that in the new *vertion* this won't be needed. Could you *confurm* this? I have spoken to *customore* support on this *be* they could not *shead* any *ligth.* I look forward to a speedy *replay* as this has been *rasied* as an important matter. There is a problem with the program *were my is* can lock up the handler. *Unforchanley there* will not *cominit develepment resourse* to *resove* this. Since it is not a *susupported* product there is *not garanity* that it will be *availbe* in the *futcher.* We do not have a *vertion witch* is *compatable.* I *surgest* that it is not shipped and that a plan is *draw* up. They are used to manage transfers *though* the PQR. *Fristly* I *appologize* for the delay. I have now *sponke* to the groups using the *pesent* system. We need to connect 8 PCs to a plotter and 2 *lasser printes.* In the *loger* term we would like access modems. Can we connect personal *computes* to the LAN, *preferbley with out* the use of gadgets? Can you give me an *ashorance* about the throughput? Please *explan* to me how we

can *enshore net work sequerity.* To *alow* me to *procide* with this project, I need a costing. I have *compleated* it and *payed poticular* attention to it. I *beleive* it is a cause of *consurn* as it is a *vertion* that will be *creected* soon. I have noticed *the* there is a *diferance* to be *agred* but *unforchantly* he has *allready surgested* that a file is *writen* to ask people for *there* name and *perpose* when *loging* in. This *infomation* could be *printer* out and a *simular* system used for the *minites.* No action has been taken on dumps *witch* are wrong but if they *which* to change the *approch,* we will need more *comlecated* plans for *suchg occurancs.* It was *explaned* and *ecpeted* that this was a *fause enviment* but the best that could be *accived.* I *beleve* that if tests are *carrey* out, it is *unlikly* this will be *repduced. Form* this reason, a new one, with lower *proformance,* was used. Please *arange* a date, by *preferance* as soon as *posible.* This will supplied but it *dose* not give *technacal resons* as they are *basced* on the patches *need* by the software. *With out consaulting* Jim I spoke to *Cris* and *surgested* a letter *explaning* it in *writting.* This *procdure* is *incompatable* with *managment.* An *assumtion* is that it is a *seperate* area and not a *swithcing* area with a *funtion.* The *configeration* should be *copyed* with the old one *runing* and *continue untill* all the old *calbling* is done. No time is *shedualed* to specify the *conferation* at the *softwere* level. I list my *perseption* of *there* status. Has this been *implemeted?* Can they supply more *infomation?* It has low *priorty. Jhon* will supply it. Fred is *monerting* the position. I try, with no *sucess,* to *reperduce* the problem. Is the *procdure valided?* Can it *destory* the data? The *commard* set works. What is the *granuality? Dose* the *lenth* set in the *feild* include this? What *scaning* time is used? We must *aviod* these *constrantes* at *persent.* There *seams* to be some *confusetion. Althrow* he *rembers* the *situartion,* he is not clear on *detailes.* With regard to *deleteing* parts, could you *advice* me of the *excat* nature of the *promblem* and I will *investgate* it *imeaditly.* I have *tryed* on a number of *ocations* while *carring* out *evualtion. Throw out* this time if the problem *reocures,* it should be *carreyed* out if *posible* and should also be *enable.*

References

Alberga, C. N. 1967 'String similarity and misspellings'. *Communications of the A.C.M.* **10** (5): 302-13.

Albrow, K. H. 1972 *The English Writing System: notes towards a description*. Longman for the Schools Council, Schools Council Programme in Linguistics and English Teaching.

Allport, D. A. and Funnell, E. 1981 'Components of the mental lexicon'. *Philosophical Transactions of the Royal Society of London* **B 295**: 397-410.

Angell, R. C., Freund, G. E. and Willett, P. 1983 'Automatic spelling correction using a trigram similarity measure'. *Information Processing and Management* **19** (4): 255-61.

Atwell, E. 1983 'Constituent-likelihood grammar'. *ICAME News: newsletter of the International Computer Archive of Modern English* (7): 34-67.

Atwell, E. and Elliott, S. 1987 'Dealing with ill-formed English text'. In Garside, Leech and Sampson (eds), *The Computational Analysis of English: a corpus-based approach*. Longman, pp. 120-38.

Baker, R. G. 1980 'Orthographic awareness'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 51-68.

Baron, J., Treiman, R., Wilf, J. F. and Kellman, P. 1980 'Spelling and reading by rules'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 159-94.

Barry, C. and Seymour, P. H. K. 1988 'Lexical priming and sound-to-spelling contingency effects in nonword spelling'. *Quarterly Journal of Experimental Psychology* **40A** (1): 5-40.

Baugh, A. C. and Cable, T. 1978 *A History of the English Language*. Routledge and Kegan Paul, Third edition.

Bellman, R. 1957 *Dynamic Programming*. Princeton University Press.

Blair, C. R. 1960 'A program for correcting spelling errors'. *Information and Control* **3**: 60-7.

Blake, N. 1992 'Introduction'. In Norman Blake (ed.), *The Cambridge History of the English Language, Volume 2*. Cambridge University Press, pp. 1-22.

Boiarsky, C. 1969 'Consistency of spelling and pronunciation deviations of Appalachian students'. *The Modern Language Journal* **53** (2): 347-50.

Booth, B. 1987 'Text input and pre-processing: Dealing with the orthographic form of texts'. In Garside, Leech and Sampson (eds), *The Computational Analysis of English: a corpus-based approach*. Longman, pp. 97-109.

Brooks, G., Gorman, T. and Kendall, L. 1993 *Spelling it out: the spelling abilities of 11- and 15-year-olds*. National Foundation for Educational Research.

Brown, A. 1986 'The pedagogical importance of consonantal features of the English of Malaysia and Singapore'. *RELC Journal* **17** (2): 1-25.

Brown, G. D. A. and Ellis, N. C. (eds) 1994 *Handbook of Spelling*. John Wiley and Sons.

Brown, H. D. 1970 'Categories of spelling difficulty in speakers of English as a first and second language'. *Journal of Verbal Learning and Verbal Behaviour* **9**: 232-6.

Bryant, P. E. and Bradley, L. 1980 'Why children sometimes write words which they do not read'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 355-70.

Burchfield, R. 1985 *The English Language*. Oxford University Press.

Burden, V. 1992 'Why are some "normal" readers such poor spellers?'. In Sterling and Robson (eds), *Psychology, Spelling and Education*. Multilingual Matters Ltd, pp. 200-14.

Campbell, R. 1983 'Writing nonwords to dictation'. *Brain and Language* **19**: 153-78.

Campbell, R. 1987 'One or two lexicons for reading and writing words: can misspellings shed any light?'. *Cognitive Neuropsychology* **4** (4): 487-99.

Carney, E. 1994 *A Survey of English Spelling*. Routledge.

Carroll, J. B., Davies, P. and Richman, B. 1971 *Word Frequency Book*. American Heritage.

Carter, D. M. 1992 'Lattice-based word identification in CLARE'. In *Proceedings of the 30th annual meeting of the A.C.L., Newark, Delaware*. Association for Computational Linguistics, pp. 159-66.

Chomsky, C. 1970 'Reading, writing and phonology'. *Harvard Educational Review* **40**: 287-309.

Chomsky, N. and Halle, M. 1968 *The Sound Pattern of English*. Harper and Row.

Church, K. W. and Gale, W. A. 1991 'Probability scoring for spelling correction'. *Statistics and Computing* **1**: 93-103.

Clark, C. 1992 'Onomastics'. In Norman Blake (ed.), *The Cambridge History of the English Language, Volume 2*. Cambridge University Press, pp. 542-606.

Cooper, W. E. 1983 'Introduction'. In William E. Cooper (ed.), *Cognitive Aspects of Skilled Typewriting*. Springer-Verlag, pp. 1-38.

Cornew, R. W. 1968 'A statistical method of spelling correction'. *Information and Control* **12**: 79-93.

Damerau, F. J. 1964 'A technique for computer detection and correction of

spelling errors'. *Communications of the A.C.M.* **7**: 171-6.

Damerau, F. J. and Mays, E. 1989 'An examination of undetected typing errors'. *Information Processing and Management* **25** (6): 659-64.

Davidson, L. 1962 'Retrieval of misspelled names in an airlines passenger record system'. *Communications of the A.C.M.* **5**: 169-71.

Desberg, P., Elliott, D. E. and Marsh, G. 1980 'American Black English and spelling'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 69-82.

Dewey, G. 1971 *English spelling: roadblock to reading*. New York Teachers' College Press.

Dougal, D. 1991 *Testing the coverage of a machine-usable standard English dictionary*. Birkbeck College, University of London, MSc Computing Science project report.

Du, M. W. and Chang, S. C. 1992 'A model and a fast algorithm for multiple errors spelling correction'. *Acta Informatica* **29**: 281-302.

Durham, I., Lamb, D. A. and Saxe, J. B. 1983 'Spelling correction in user interfaces'. *Communications of the A.C.M.* **26** (10): 764-73.

Ehri, L. C. 1980 'The development of orthographic images'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 311-38.

Ellis, A. W. 1979 'Slips of the pen'. *Visible Language* **13** (3): 265-82.

Ellis, A. W. 1984 *Reading, Writing and Dyslexia: A Cognitive Analysis*. Lawrence Erlbaum Associates.

Ellis, N. C. 1994 'Longitudinal studies of spelling development'. In Brown and Ellis (eds), *Handbook of Spelling*. John Wiley and Sons, pp. 155-78.

Ellis, N. and Cataldo, S. 1992 'Spelling is integral to learning to read'. In Sterling and Robson (eds), *Psychology, Spelling and Education*. Multilingual Matters Ltd, pp. 122-42.

Ewert, A. 1933 *The French Language*. Faber and Faber.

Fisher, J. H., Richardson, M. and Fisher, J. L. 1984 *An Anthology of Chancery English*. University of Tennessee Press.

Follick, M. 1965 *The Case for Spelling Reform*. Sir Isaac Pitman and Sons.

Forney, G. D. 1973 'The Viterbi Algorithm'. *Proc IEEE* **61** (3): 268-78.

Fowler, H. W. 1968 *A Dictionary of Modern English Usage*. Oxford University Press, Second edition, revised by Sir Ernest Gowers.

Francis, W. N. and Kučera, H. 1982 *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin.

Frith, U. 1980 'Unexpected spelling problems'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 495-515.

Frith, U. 1986 'A developmental framework for developmental dyslexia'. *Annals of Dyslexia* **36**: 69-81.

Funnell, E. 1992 'On recognising misspelled words'. In Sterling and Robson (eds), *Psychology, Spelling and Education*. Multilingual Matters Ltd, pp. 87-99.

Gale, W. A. and Church, K. W. 1990 'Estimation procedures for language context: poor estimates are worse than none'. In K. Momirovic and V. Mildner (eds), *Compstat 1990, Proceedings in Computational Statistics*. Physica-Verlag Heidelberg for International Association for Statistical Computing, pp. 69-74, 9th symposium held at Dubrovnik, Yugoslavia.

Garside, R., Leech, G. and Sampson, G. 1987 *The Computational Analysis of English: a corpus-based approach*. Longman.

Gentner, D. R. 1983 'Keystroke timing in transcription typing'. In William E. Cooper (ed.), *Cognitive Aspects of Skilled Typewriting*. Springer-Verlag, pp. 95-120.

Gentner, D. R., Larochelle, S. and Grudin, J. 1988 'Lexical, sublexical and peripheral effects in skilled typewriting'. *Cognitive Psychology* **20**: 524-48.

Gleason, H. A. 1961 *An Introduction to Descriptive Linguistics*. Holt, Rinehart and Winston.

Gorman, T. P. 1981 'A survey of attainment and progress of learners in adult literacy schemes'. *Educational Research* **23** (3): 190-8.

Goswami, U. and Bryant, P. 1990 *Phonological Skills and Learning to Read*. Lawrence Erlbaum Associates Ltd..

Goulandris, N. K. 1992 'Alphabetic spelling: predicting eventual literacy attainment'. In Sterling and Robson (eds), *Psychology, Spelling and Education*. Multilingual Matters Ltd, pp. 143-58.

Graham, R. T. and Rudorf, E. H. 1970 'Dialect and spelling'. *Elementary English* **47** (3): 363-76.

Griswold, R. E. and Griswold, M. T. 1983 *The Icon Programming Language*. Prentice-Hall.

Grudin, J. T. 1983 'Error patterns in novice and skilled transcription typing'. In William E. Cooper (ed.), *Cognitive Aspects of Skilled Typewriting*. Springer-Verlag, pp. 121-44.

Haas, W. 1969 'On "spelling" and "spelling reform"'. In W. Haas (ed.), *Alphabets for English*. Manchester University Press, pp. 1-13.

Haas, W. 1970 *Phono-graphic Translation*. Manchester University Press, Mont Follick Series Vol 2.

Hall, P. A. V. and Dowling, G. R. 1980 'Approximate string matching'. *Computing Surveys* **12** (4): 381-402.

Hamilton, M. and Stasinopoulos, M. 1987 *Literacy, Numeracy and Adults*. Adult Literacy and Basic Skills Unit.

Hanks, P. 1988 'Conventionality and efficiency in written English: the hyphen'. *Journal of the Simplified Spelling Society* **2** (2): 5-10.

Hanna, P. R., Hanna, J. S., Hodges, R. E. and Rudorf, E. H. J. 1966 *Phoneme-grapheme correspondences as cues to spelling improvement*. U.S. Government Printing Office.

Heidorn, G. E., Jensen, K., Miller, L. A., Byrd, R. J. and Chodorow, M. S. 1982 'The EPISTLE text-critiquing system'. *IBM Systems Journal* **21** (3): 305-26.

Henderson, L. and Chard, J. 1980 'The reader's implicit knowledge of

orthographic structure'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 85-116.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D. and Slocum, J. 1978 'Developing a natural language interface to complex data'. *A.C.M. Transactions on Database Systems* **3** (2): 105-47.

Hofland, K. and Johansson, S. 1982 *Word Frequencies in British and American English*. Norwegian Computing Centre for the Humanities/ Longman.

Hogg, R. M. 1992 'Phonology and Morphology'. In Richard M. Hogg (ed.), *The Cambridge History of the English Language, Volume 1*. Cambridge University Press, pp. 67-167.

Holbrook, D. 1964 *English for the Rejected*. Cambridge University Press.

Holmes, V. M. and Ng, E. 1993 'Word-specific knowledge, word-recognition strategies and spelling ability'. *Journal of Memory and Language* **32**: 230-57.

Hotopf, N. 1980 'Slips of the pen'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 287-307.

Houghton, G., Glasspool, D. W. and Shallice, T. 1994 'Spelling and serial recall: insights from a competitive queueing model'. In Brown and Ellis (eds), *Handbook of Spelling*. John Wiley and Sons, pp. 365-404.

Huxford, L., Terrell, C. and Bradley, L. 1992 '"Invented" spelling and learning to read'. In Sterling and Robson (eds), *Psychology, Spelling and Education*. Multilingual Matters Ltd, pp. 159-67.

Innes, P. 1990 *Defeating Dyslexia: a boy's story*. Kyle Cathie Ltd.

Jaquith, J. R. 1976 'Digraphia in advertising: the public as guinea pig'. *Visible Language* **10** (4): 295-308.

Johansson, S. 1986 *The Tagged LOB Corpus: User's Manual*. Norwegian Computing Centre for the Humanities.

Johnson, S. 1755 *A Dictionary of the English Language*. London.

Joseph, D. M. and Wong, R. L. 1979 'Correction of misspellings and typographical errors in a free-text medical English information storage and retrieval system'. *Methods of Information in Medicine* **18** (4): 228-34.

Kempen, G. and Vosse, T. 1992 'A language-sensitive text editor for Dutch'. In Patrick O'Brian Holt (ed.), *Computers and Writing*. Intellect Books, pp. 68-77.

Kernighan, M. D., Church, K. W. and Gale, W. A. 1990 'A spelling correction program based on a noisy channel model'. In Hans Karlgren (ed.), *COLING-90 13th International Conference on Computational Linguistics*. Helsinki University, pp. 205-10, Volume 2.

Kingman, J. 1988 *Report of the Committee of Inquiry into the teaching of English Language (The Kingman Report)*. H.M.S.O., Department of Education and Science.

Knuth, D. E. 1973 *The Art of Computer Programming: Volume 3 Sorting and Searching*. Addison-Wesley.

Kučera, H. and Francis, W. N. 1967 *Computational Analysis of Present-day*

American English. Brown University Press.

Kukich, K. 1992a 'Techniques for automatically correcting words in text'. *Computing Surveys* **24** (4): 377-439.

Kukich, K. 1992b 'Spelling correction for the Telecommunications Network for the Deaf'. *Communications of the A.C.M.* **35** (5): 80-90.

Lass, R. 1992 'Phonology and Morphology'. In Norman Blake (ed.), *The Cambridge History of the English Language, Volume 2*. Cambridge University Press, pp. 23-155.

Leech, G. N., Garside, R. G. and Elliott, S. J. 1986 *Development of a Context-sensitive Textual Error Detector and Corrector: Final project report submitted to International Computers Limited*. Unit for Computer Research on the English Language, Lancaster University.

Leech, G. 1993 '100 million words of English; the British National Corpus project'. *English Today* **9** (1): 9-16, ET33.

Lennox, C. and Siegel, L. S. 1994 'The role of phonological and orthographic processes in learning to spell'. In Brown and Ellis (eds), *Handbook of Spelling*. John Wiley and Sons, pp. 93-110.

Lesk, M. 1987 *Automatic sense disambiguation: how to tell a pine cone from an ice cream cone*. Bell Communications Research, Research report.

Levenshtein, V. I. 1966 'Binary codes capable of correcting deletions, insertions and reversals'. *Soviet Physics - Doklady* **10** (8): 707-10.

Long, J. 1976 'Visual feedback and skilled keying: differential effects of masking the printed copy and the keyboard'. *Ergonomics* **19**: 93-110.

Lowrance, R. and Wagner, R. A. 1975 'An extension of the string-to-string correction problem'. *Journal of the A.C.M.* **22** (2): 177-83.

MacCarthy, P. A. D. 1969a 'New spelling with old letters'. In W. Haas (ed.), *Alphabets for English*. Manchester University Press, pp. 89-104.

MacCarthy, P. A. D. 1969b 'The Bernard Shaw Alphabet'. In W. Haas (ed.), *Alphabets for English*. Manchester University Press, pp. 105-17.

Marcel, T. 1980 'Phonological awareness and phonological representation: investigation of a specific spelling problem'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 373-403.

Marshall, I. 1983 'Choice of grammatical word-class without global syntactic analysis: tagging words in the LOB Corpus'. *Computers and the Humanities* **17**: 139-50.

Masterman, M. 1979 'Rhetorical punctuation by machine'. In D.E. Ager, F.E. Knowles, J. Smith (eds), *Advances in Computer-aided Literary and Linguistic Research*., pp. 289-320, Proceedings of the Fifth International Symposium on Computers in Literary and Linguistic Research, Aston University.

Mays, E., Damerau, F. J. and Mercer, R. L. 1991 'Context based spelling correction'. *Information Processing and Management* **27** (5): 517-22.

McIlroy, M. D. 1982 'Development of a spelling list'. *IEEE Transactions on Communications* **COM-30** (1): 91-9.

McLeod, P. and Hume, M. 1994 'Overlapping mental operations in serial performance with preview: typing - a reply to Pashler'. *Quarterly Journal of Experimental Psychology* **47A** (1): 193-9.

Milroy, J. 1992 'Middle English Dialectology'. In Norman Blake (ed.), *The Cambridge History of the English Language, Volume 2*. Cambridge University Press, pp. 156-206.

Mitton, R. 1985 'A collection of computer-readable corpora of English spelling errors'. *Cognitive Neuropsychology* **2** (3): 275-9.

Mitton, R. 1986 'A partial dictionary of English in computer-usable form'. *Literary and Linguistic Computing* **1** (4): 214-5.

Mitton, R. 1987 'Spelling checkers, spelling correctors and the misspellings of poor spellers'. *Information Processing and Management* **23** (5): 495-505.

Moon, R. 1987 'The analysis of meaning'. In J.M. Sinclair (ed.), *Looking Up: an account of the COBUILD project in lexical computing*. Collins, pp. 86-103.

Morgan, H. L. 1970 'Spelling correction in systems programs'. *Communications of the A.C.M.* **13** (2): 90-4.

Morris, R. and Cherry, L. L. 1975 'Computer detection of typographical errors'. *IEEE Trans Professional Communication* **PC-18** (1): 54-64.

Morton, J. 1980 'The logogen model and orthographic structure'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 117-33.

Moseley, D. 1989 'How lack of confidence in spelling affects children's written expression'. *Educational Psychology in Practice* **5** (1): 42-6.

Nix, R. 1981 'Experience with a space efficient way to store a dictionary'. *Communications of the A.C.M.* **24** (5): 297-8.

Norman, D. A. and Rumelhart, D. E. 1983 'Studies of typing from the LNR research group'. In William E. Cooper (ed.), *Cognitive Aspects of Skilled Typewriting*. Springer-Verlag, pp. 45-65.

Okuda, T., Tanaka, E. and Kasai, T. 1976 'A method for the correction of garbled words based on the Levenshtein metric'. *IEEE Transactions on Computers* **C-25** (2): 172-8.

Perin, D. 1983 'Phonemic segmentation and spelling'. *British Journal of Psychology* **74**: 129-44.

Peters, M. L. 1970 *Success in Spelling: a study of the factors affecting improvement in spelling in the junior school*. Cambridge Institute of Education.

Peterson, J. L. 1980a 'Computer programs for detecting and correcting spelling errors'. *Communications of the A.C.M.* **23** (12): 676-87.

Peterson, J. L. 1980b *Computer Programs for Spelling Correction: an experiment in program design*. Springer-Verlag, Lecture Notes in Computer Science: 96.

Peterson, J. L. 1986 'A note on undetected typing errors'. *Communications of the A.C.M.* **29** (7): 633-7.

Pitman, J. 1969 'The late Dr Mont Follick - an appraisal: the assault on the conventional alphabets and spelling'. In W. Haas (ed.), *Alphabets for English*.

Manchester University Press, pp. 14-49.

Pitman, J. and St John, J. 1969 *Alphabets and Reading*. Sir Isaac Pitman and Sons Ltd.

Pollock, J. J. and Zamora, A. 1983 'Collection and characterization of spelling errors in scientific and scholarly text'. *Journal of the American Society for Information Science* **34** (1): 51-8.

Pollock, J. J. and Zamora, A. 1984 'Automatic spelling correction in scientific and scholarly text'. *Communications of the A.C.M.* **27** (4): 358-68.

Rabbitt, P. 1978 'Detection of errors by skilled typists'. *Ergonomics* **21** (11): 945-58.

Rabiner, L. R. and Juang, B. H. 1986 'An introduction to Hidden Markov Models'. *IEEE ASSP (Acoustics Speech and Signal Processing) Magazine* **3** (1): 4-16.

Ramshaw, L. A. 1994 'Correcting real-word spelling errors using a model of the problem-solving context'. *Computational Intelligence* **10** (2): 185-211.

Read, C. 1973 'Children's judgments of phonetic similarities in relation to English spelling'. *Language Learning* **23** (1): 17-38.

Richardson, S. D. 1985 *Enhanced Text Critiquing using a Natural Language Parser*. IBM, Research Report RC 11332 (#51041).

Ripman, W. and Archer, W. 1940 *New Spelling*. Pitman.

Riseman, E. M. and Hanson, A. R. 1974 'A contextual post-processing system for error correction using binary n-grams'. *IEEE Trans Computers* **C-23** (5): 480-93.

Rodgers, B. 1986 'Change in the reading attainment of adults: a longitudinal study'. *British Journal of Developmental Psychology* **4**: 1-17.

Rumelhart, D. E. and Norman, D. A. 1982 'Simulating a skilled typist: a study of skilled motor performance'. *Cognitive Science* **6**: 1-36.

Salmon, V. forthcoming 'Orthography and Punctuation'. In Roger Lass (ed.), *The Cambridge History of the English Language, Volume 3*. Cambridge University Press.

Salthouse, T. A. 1984 'The skill of typing'. *Scientific American* **250** (2): 94-9.

Salthouse, T. A. 1986 'Perceptual, cognitive and motoric aspects of transcription typing'. *Psychological Bulletin* **99** (3): 303-19.

Sampson, G. 1985 *Writing Systems*. Hutchinson.

Sampson, G. 1989 'How fully does a machine-usable dictionary cover English text?'. *Literary and Linguistic Computing* **4** (1): 29-35.

Samuels, M. L. 1972 *Linguistic Evolution*. Cambridge University Press, Cambridge Studies in Linguistics 5.

Scragg, D. G. 1974 *A History of English Spelling*. Manchester University Press, Mont Follick Series Vol 3.

Scudder, H. E. 1882 *Noah Webster*. Houghton, Mifflin and company.

Seymour, P. H. K. and Dargie, A. 1990 'Associative priming and orthographic choice in nonword spelling'. *European Journal of Cognitive Psychology* **2**: 395-410.

Shaffer, L. H. 1975 'Control processes in typing'. *Quarterly Journal of Experimental Psychology* **27** (3): 419-32.

Shaffer, L. H. and Hardwick, J. 1968 'Typing performance as a function of text'. *Quarterly Journal of Experimental Psychology* **20** (4): 360-72.

Shaffer, L. H. and Hardwick, J. 1969 'Errors and error detection in typing'. *Quarterly Journal of Experimental Psychology* **21** (3): 209-13.

Shallice, T. 1981 'Phonological agraphia and the lexical route in writing'. *Brain* **104**: 413-29.

Shaw, G. B. 1962 *Androcles and the Lion*. Penguin Books, Shaw alphabet edition.

Simon, D. P. and Simon, H. A. 1973 'Alternative uses of phonemic information in spelling'. *Review of Educational Research* **43**: 115-37.

Sinclair, J. M. (ed.) 1987 *Looking Up: an account of the COBUILD project in lexical computing*. Collins.

Singleton, C. (ed.) 1991 *Computers and Literacy Skills*. British Dyslexia Association Computer Resource Centre.

Skousen, R. 1982 'English spelling and phonemic representation'. *Visible Language* **16** (1): 28-38.

Sloboda, J. A. 1980 'Visual imagery and individual differences in spelling'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 231-48.

Smith, P. T. 1980a 'In defence of conservatism in English orthography'. *Visible Language* **14** (2): 122-36.

Smith, P. T. 1980b 'Linguistic information in spelling'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 33-49.

Smith, P. T. 1983 'Patterns of writing errors in the framework of an information-processing model of writing'. In Rogers and Sloboda (eds), *The Acquisition of Symbolic Skills*. Plenum Press, pp. 171-7.

Snowling, M. J. 1994 'Towards a model of spelling acquisition: the development of some component skills'. In Brown and Ellis (eds), *Handbook of Spelling*. John Wiley and Sons, pp. 111-28.

Stanovich, K. E. and West, R. F. 1989 'Exposure to print and orthographic processing'. *Reading Research Quarterly* **24** (1): 402-33.

Sterling, C. M. 1983 'Spelling errors in context'. *British Journal of Psychology* **74**: 353-64.

Sterling, C. M. and Robson, C. (eds) 1992 *Psychology, Spelling and Education*. Multilingual Matters Ltd.

Strang, B. M. H. 1970 *A History of English*. Methuen.

Sun, W., Liu, L., Zhang, W. and Comfort, J. C. 1992 'Intelligent OCR processing'. *Journal of the American Society for Information Science* **43** (6): 422-31.

Sweet, H. 1876 'Words, logic and grammar'. *Transactions of the Philological Society*.

Teitelman, W. 1972 '"Do What I Mean": the programmer's assistant'. *Computers and Automation*: 8-11.

Tenney, Y. J. 1980 'Visual factors in spelling'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 215-29.

Thorndike, E. L. and Lorge, I. 1944 *The Teacher's Word Book of 30,000 Words*. Teachers College, Columbia University.

Treiman, R. 1994 'Sources of information used by beginning spellers'. In Brown and Ellis (eds), *Handbook of Spelling*. John Wiley and Sons, pp. 75-92.

Turba, T. N. 1982 'Length-segmented lists'. *Communications of the A.C.M.* **25** (8): 522-6.

Upward, C. 1988 *English Spelling and Educational Progress*. British Association for Applied Linguistics Committee for Linguistics in Education, CLIE Working Paper Number 11.

Upward, C. 1992 *Cut Spelling*. Simplified Spelling Society.

Upward, C. 1994 'Err analysis: som reflections on aims, methods, limitations and importnce, with a furthr demnstration: Part 1'. *Journal of the Simplified Spelling Society* (1): 29-33.

Vachek, J. 1973 *Written Language: general problems and problems of English*. Mouton, The Hague, Janua Linguarum Series Critica 14.

Vaidya, S. 1992 *The effect of collocational information on spelling correction*. Birkbeck College, University of London, MSc Computing Science project report.

Van Berkel, B. and De Smedt, K. 1988 'Triphone analysis: a combined method for the correction of orthographical and typographical errors'. In *Second Conference in Applied Natural Language Processing, Austin, Texas*. American Association of Computational Linguistics, pp. 77-83.

Venezky, R. L. 1970 *The Structure of English Orthography*. Mouton, The Hague, Janua Linguarum No 82.

Venezky, R. L. 1976 'Notes on the history of English spelling'. *Visible Language* **10** (4): 351-65.

Venezky, R. L. 1980 'From Webster to Rice to Roosevelt'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 9-30.

Veronis, J. 1988 'Computerized correction of phonographic errors'. *Computers and the Humanities* **22**: 43-56.

Wagner, R. A. and Fischer, M. J. 1974 'The string-to-string correction problem'. *Journal of the A.C.M.* **21** (1): 168-73.

Warburton, F. W. and Southgate, V. 1969 *i.t.a. : An Independent Evaluation*. John Murray and Chambers for The Schools Council.

Wells, J. C. 1986 'English accents and their implications for spelling reform'. *Simplified Spelling Society Newsletter* (2): 5-13.

Wells, J. C. 1982 *Accents of English*. Cambridge University Press.

Wijk, A. 1966 *Rules of Pronunciation for the English Language*. Oxford University Press.

Wijk, A. 1969 'Regularized English'. In W. Haas (ed.), *Alphabets for English*. Manchester University Press, pp. 50-88.

Wing, A. M. and Baddeley, A. D. 1980 'Spelling errors in handwriting: a corpus

and a distributional analysis'. In Uta Frith (ed.), *Cognitive Processes in Spelling*. Academic Press, pp. 251-85.

Woods, W. A. 1970 'Transition network grammars for natural language analysis'. *Communications of the Association for Computing Machinery* **13** (10): 591-606.

Wrenn, C. L. 1949 *The English Language*. Methuen.

Yannakoudakis, E. J. and Fawthrop, D. 1983a 'The rules of spelling errors'. *Information Processing and Management* **19** (2): 87-99.

Yannakoudakis, E. J. and Fawthrop, D. 1983b 'An intelligent spelling error corrector'. *Information Processing and Management* **19** (2): 101-8.

Yianilos, P. N. 1983 'A dedicated comparator matches symbol strings fast and intelligently'. *Electronics*: 113-7.

Yule, V. 1978 'Is there evidence for Chomsky's interpretation of English spelling?'. *Spelling Progress Bulletin* **18** (4): 10-2.

Index