Justification with fewer hyphens

Alison M. Pringle

University of Cambridge Computer Laboratory, Corn Exchange Street, Cambridge CB2 3QG, UK

Many books are now typeset by computer, and certain areas of this work are subject to much criticism by professional printers. An example is justification and the introduction of hyphenation. Many of the programs used in the past have been naive and inflexible. This paper presents an approach to the problem which is both new and flexible, and produces better results than many which have been seen before now. In some cases it may even produce results which improve on those of traditional methods.

(Received September 1980)

Introduction

The application area within which this paper falls is that of the computer based composition of text. This is an area which has expanded rapidly in the last ten years. Not only has the number of available systems expanded enormously, but the measure of acceptance and approval accorded to them by the printing industry has increased. This approval is not unqualified and there are several areas where the work produced by these means is much criticised. This paper is an account of one of a group of experiments which were designed to demonstrate that in some areas much better results can be obtained from automated systems than have been seen in the past, and that these improved results may in some cases make available possibilities which were not readily available through traditional methods.

There is one point which must be emphasised before detailed consideration of the work described here can begin: the type of printing work in which we are interested. The discussion here applies to high quality work of the sort produced by printers for bookwork. It does not specifically apply to the kind of line-printer paginator with which most computer scientists are familiar or to the 'near print' which some word processing systems emit. It may well be that some of the principles are applicable in that context, although they might be regarded as unduly fussy, but it is not our target area. The work is aimed at text which will be properly printed to commercial standard by professional printers.

The particular problem under consideration in this case is that of justification. Justification is the process of breaking text into lines. It consists of choosing line endings and altering the spaces between words and at each end of each line in such a way that the text forms a regular and pleasing pattern on the page. This is usually interpreted as meaning that the right-hand margin of printed text should form a straight line (right justification). In fact similar considerations apply however text is set. For example, it may be 'set left', that is with a straight left-hand but ragged right-hand margin. It may also be centred or set in various non-standard ways, as poetry often is, but analogous problems still occur. To some extent it is true that the problems are more acute for right-justified text. For this reason (and because this is the most frequently used format) it is to this particular problem that the rest of this section applies in detail. In general there are limits to the range of values suitable for word spacing. If words are too closely packed it will be difficult to distinguish them, while if they are too loose the flow of the text is lost, as is the strong horizontal guide needed by the eye. Somewhere between the minimum and maximum acceptable spaces is an ideal space. This may not, in fact, bisect the range. For example, a typical range might be four to nine printers' units with a standard space of six units (a printers' unit is usually defined as one eighteenth of the width of an upper case 'M' in the current type, or body, size and is thus scaled with the body size). An alternative method would be to distribute space variously through the text, between letters as well as words. This is not generally well thought of by printers, and is not considered here. Yet another approach would be to use the freedom provided by modern photosetters to scale characters independently in the horizontal and vertical directions. This would receive even less approval in principle than the last suggestion, although experiment suggests that people are in fact less conscious of the visual effects than they believe if it is done with discretion. However, this too is not considered here.

The usual approach to justification consists of: filling the current line with words until it is full or overfull; determining which words (if any) will lead to acceptable spacing if chosen as the line end; and selecting the one which gives a spacing closest to the ideal. If no such line break can be found, select the word whose ends span the possible space range and hyphenate it in such a way that an acceptable word spacing results.

Although this may seem a reasonable approach and has been used with some success in many systems, it does have drawbacks. Since each line is considered separately the overall effect may be unsatisfactory. A typical rule for word division is given in Hart's Rules (1978, p. 137), 'Two successive hyphens only are allowed at the end of lines'. **Fig. 1** shows the bottom section of the first page of text from a directory (CIT, 1978) which was set by computer. This violates not only this rule, but also the next: 'A divided word should not end a right-hand page'. It is also considered bad practice to finish a paragraph with a line consisting only of the second part of a broken word.

A further difficulty lies in the hyphenation itself. English is a very difficult language to hyphenate correctly, since almost any general rule has exceptions, e.g. 'mo-ther' but 'light-house'.This demonstrates why algorithms such as those described below will often fail. There are also well-known hazardous words, such as 'therapist'. In addition there are semantic problems which cannot be resolved by inspecting letter patterns. A good example is 'present' which is split 'pre-sent' or 'pres-ent' according to the sense in which it is employed. Printers themselves are acutely aware of this problem, and many books have been published about hyphenation, for example Sisam (1929). Many algorithms for automated hyphenation have also been published [e.g. Rich and Stone (1965); Ocker (1975); Knuth (1979); Moitra *et al.* (1979)].

There are two main techniques which have been used for hyphenation in automated systems. The first is the *logic hyphenator*. In this case an algorithm is provided which searches a word for syllables, and hence hyphenation points. A common

CCC-0010-4620/81/0024-0320 $02.00

What is the Convocation?

by E.F. LAWLOR

Chairman of the Standing Committee of the Convocation

Nearly five thousand graduates have passed through the Cranfield Institute of Technology and its predecessor, the College of Aeronautics, and together they make up most of the membership of Convocation, which is a constituent body of the Institute. This body is frequently called 'The Cranfield Society', a name inherited from the association of former students of the College of Aeronautics.

From its earliest days the Society was aware of its potential for highlighting key issues, and during the fifties and sixties this developed in several ways. The Society held a number of weekend Symposia, devoted to discussions of future trends in Aeronautics, Education, Communications, Transport and Management. During this period the Society prepared submissions to the Plowden Committee on the Aircraft Industry, the Fulton Committee on the Civil Service and the Board of Trade Civil Aviation Enquiry. Another activity was a series of annual lectures, organised in collaboration with the Royal Aeronautical Society, in memory of the late Sir Frederick Handley Page, and having the theme 'The Influence of Aviation and Astronautics on Human Affairs'. We were honoured to have HRH The Duke of Edinburgh, KG deliver the inaugural lecture at Cranfield in 1963. Since the early sixties a prize has been awarded each year to the student covering the widest range of achievement, both socially and academically.

Towards the end of the sixties, the Society turned its attention more and more to the role it could usefully adopt within the Cranfield Institute of Technology. When the Institute received its Charter at the end of 1969 it provided for a Convocation of graduates, whose prime function was to bring to bear their interest, experience and views towards achievement of the objects of the Institute. This is the role which the Convocation has set out to fulfil since the Charter was granted.

But this is not its only role. It also seeks to provide fellowship between members by organising functions such as dinners, dances and social gatherings. In addition it publishes its own news magazine *Digest* twice a year. In order to have the opportunity to participate in these activities, and to receive *Digest*, all members are invited to pay a fee (currently £2 p.a. or £40 for life). Those who do not become 'participating members' in this way normally receive information only once a year concerning the Annual General Meeting. However, every member (whether participating or not) has been sent a copy of this Directory if we have his or her address.

Fig. 1 Example of computer typesetting

method is to start at the end of the word and scan forward until a vowel is found. This indicates the centre of a syllable. Since in English the preceding consonant is normally part of the syllable, the scan is continued until a consonant is found. This is then considered jointly with the preceding letter, to see if they form a normally unbreakable pair, such as 'th' or 'gh', and the syllable start chosen accordingly. Several such algorithms have been published, and many systems have been written using them. The algorithm published by Rich and Stone (1965) is a good example of a pure algorithm of this type. The difficulty with this approach is that however sophisticated such algorithms are, they can rarely give better than 60–70% correct hyphenation. It can, however, be argued that such an algorithm is quick, and thus comparatively cheap, and that as long as all hyphenations are recorded by the system and checked by a reader it is acceptable to correct bad ones manually—treating them like spelling mistakes in other parts of the text.

An alternative approach is to keep a dictionary of words longer than (say) five letters, together with their possible break points. All hyphenations are then looked up as and when they are required. The argument for using this approach is that hyphenation is, or should be, a comparatively rare event, and thus the increased expense of dictionary look-up, as opposed to using an algorithmic method, is not significant. However, large amounts of backing store would be required and the method might still fail if a new word were encountered or a contentious hyphenation used.

In practice a hybrid method is often employed. In this case an exception dictionary is maintained which contains only those words for which the algorithm is known to fail. A word to be divided is checked against this list. If it is found, a breakpoint is chosen from the dictionary, otherwise an algorithmic method is employed. It is clear that even this approach requires a considerable dictionary since it needs to contain 30–40% of the possible vocabulary. A measure of the problem can be found by considering some well-known dictionaries. Chambers (1972) contains about 150000 words, while the OED (1933) gives about 550000, or 6000000 if all the variants (eat, eats, eating. . .) are counted.

Even when an adequate solution has been found for English, or at least that subset of English commonly used in books processed by the system, problems still remain if part of the text is in another language, for which the rules developed for English do not apply.

Given that hyphenation causes such trouble, it seemed reasonable to consider an experiment on how best to avoid the problem. The most obvious method might seem to be an extension of the range of acceptable spaces. In practice this is not satisfactory since it tends to erode the quality of the text by producing unacceptably loose spacing. It is also likely to be unsuccessful if certain sorts of word pattern occur within a line. For example, a line which almost contains the word

honorificabilitudinitatibus

(Shakespeare, *Love's Labour's Lost*, Act 5 Scene 1) at the end is unlikely to be assisted by this means.

It seemed clear that more interesting results might be available if a more radical approach were adopted and complete paragraphs rather than individual lines were considered. Given that the word space for each line is chosen to be as close to a standard space as possible, it may be possible to tighten or loosen a particular line and still produce an acceptable result. This may be achieved by moving a word from one end of a line to the preceding or succeding line. Now, suppose lines are made up one at a time. For a while all will probably go well, but sooner or later a line will be found for which an appropriate line break is not available at the end of a word. It may be possible to push the first word (or words) of the current line back to the previous line. This will cause the previous line to become much tighter. If it becomes too tight it may be possible to remedy the situation by moving words back to the line before that; and so on, back, if necessary, to the beginning of the paragraph. If it is not possible to reach stability by these means, it may be possible to achieve it by the reverse process. In this case, instead of pushing words back to tighten earlier lines, you pull them forward to loosen the lines. Only if justification cannot be achieved by either means do you finally employ hyphenation. This approach is particularly attractive since it is similar to one which might, in principle, be used manually by printers. In fact they do not use it since a line sent to a casting machine is committed for good; when it is punched on a Monotype D-board there is no going back; when work is being hand set, backing up is possible but is almost the same as resetting. Juggle (see later) employs this technique to make up paragraphs.

Since completing this work, we have found evidence that a similar idea was considered and discarded by Elliotts while working on the Garden City Press system in the mid-1960s. The paper in which it is discussed (Cooper, 1967) is not explicit and it is difficult to assess the results. The main reason given for abandoning the experiment is the large resources required. These no longer seem excessive in comparison with the benefits which may be obtained. Computer composition has now reached a stage where emphasis can rightly be placed on improving quality rather than simply on efficiency, since it is this aspect which raises doubts in printers' minds rather than cost.

The program Juggle

It should be noted from the outset that Juggle was always envisaged as an experiment in a particular technique. Its capabilities are limited to the composition of paragraphs according to a format supplied by the user and the output of them in a form suitable for display.

Juggle incorporates a very basic idea of a paragraph. A paragraph is regarded as a series of words distributed into lines. The lines are made up to a specific measure and the first one may be indented. Within a line the word spacing may take values within a specified range, but an optimum is always aimed at.

The program takes three input files, one of which may be omitted. The first (and optional) one is a format description. This is short and simple, and if omitted is supplied by default. The second input file specifies the characters in the fount to be used together with their widths. Finally, and most importantly, there is the file containing the text for composition. This may contain several paragraphs, each terminated by the escape sequence '*P'.

Juggle first assimilates the format description and the fount definition, and then proceeds to read a paragraph of raw text. As it does so, it builds a data structure which reflects the verbal structure of the text. Each word is stored and measured, and information about its type is recorded. Initially there are three types of word. First there are simple words consisting purely of letters. Then there are abnormal words which contain non-letters, for example '1984', 'ALGOL68' or 'Hen3ry'. These are usefully distinguished since usually they must not be divided. It is a corollary of the way Juggle looks for words that punctuation is normally attached to the word immediately preceding it. That word is thus labelled as abnormal and will not be broken which neatly results in the correct practice in most cases. Finally there are words which are already hyphenated, such as 'well-known' or 'water-colour'. These words must not have further hyphens inserted in them, but their indigenous hyphens may be used as line breaks. The parts are therefore stored as separate words, and in fact given different types according to whether or not they represent the final part of a word. There is a clear distinction between these hyphens and any which may be inserted by the program.

When the data structure is complete Juggle sorts the words into lines using the following algorithm. Words are added to the line until the word space needed for justification is less than or equal to the maximum permitted space. Further words may then be added or removed until a spacing as close as possible to the ideal is reached. It will sometimes happen that the space which terminates the first stage is not only less than the maximum but also less than the minimum permitted space. It is at this point that the novel part of the algorithm comes into play.

The first response to this situation is to attempt to make space in the present line by pushing material backward from the line start in such a way that the earlier lines are tightened within acceptable limits. The routine which does this (PUSHBACK) is recursive. It attempts to fit the first word of the line on which trouble occurred on to the previous line. If it will fit, all well and good. Otherwise PUSHBACK calls itself in an attempt to push material from the line on which it is operating yet further back. At any point in this process where PUSHBACK is called it may be called repetitively, and a count will be maintained. Thus if an attempt to create space at the start of a line is successful but inadequate, a second attempt will be made, and so on.

If the attempt to gain space by pushing material back is unsuccessful, the alternative approach is tried. First the paragraph is restored to its state prior to the application of PUSHBACK, and then an inverse routine, PULLON, is applied. This is also, incidentally, the means used to restore the status quo. PULLON acts as an inverse to PUSHBACK, and an equal number of applications will reverse the results of applying PUSHBACK. The object of applying PULLON is to add sufficient material at the start of the problem line to make an earlier potential line break acceptable. PULLON is also recursive and acts in a manner precisely similar to PUSHBACK.

If this attempt is also unsuccessful in producing a suitable line break and word space, then the paragraph is again restored to its initial state. A logic hyphenator may then be applied. The application of the hyphenator has the side effect of modifying the data structure describing the verbal structure of the paragraph, since one word must now be represented by two. It also requires the addition of two new word types analogous to those used for naturally divided words.

When the entire paragraph has been processed in this way, it is output as linear text with appropriate typesetting directives embedded in it, and the next paragraph (if any) is processed.

The program was written in BCPL and implemented on the University of Cambridge IBM 370/165. It consists of about 8K of code, including extensive tracing code and the standard BCPL library and diagnostic aids.

Experience and results

Fig. 2 shows a paragraph generated by Juggle. This result was achieved by use of the techniques described above. Examination of trace output from the program reveals how it was done.

Our modern view of the classical era has been so much formed by the accumulated reverence for Beethoven that we accept without question the doctrine that the symphony is of all musical forms the most important, the one into which a composer must inevitably pour all his mightiest inspirations. That certainly was not the view of Beethoven's own period. The generation which had just lost Mozart and had exalted his memory to a place among the gods was inclined to regard operas and concertos as far more important than symphonies. The concerto was obviously a more important form than the symphony, because it was an occasion for watching the composer himself apparently in the very act of composing. It is difficult for us to imagine a period of musical history in which there were no classics, and in which all interest was concentrated on the newest production. The English, by commemorating Handel in Westminster Abbey in 1784, had taken the first step towards establishing a cult of 'the classics' in music, and this cult of Handel had just begun to spread from England to Germany. Johann Sebastian Bach, who for us to-day is the great classic of the eighteenth century, was practically unknown outside Leipzig. The only works of Bach which Beethoven is likely to have known were the Forty-Eight Preludes and Fugues; Forkel had just begun to awaken interest in Bach's music, and hardly any of it was accessible in print.

Fig. 2 A paragraph generated by Juggle

(a) Our modern view of the classical era has been so much formed by the accumulated....

(b) Our modern view of the classical era has been so much formed by the accumulated reverence for Beethoven that we accept without question

(c) Our modern view of the classical era has been so much formed by the accumulated reverence for Beethoven that we accept without

(d) Our modern view of the classical era has been so much formed by the accumulated reverence for Beethoven that we accept without question the doctrine....

Fig. 3 Examination of the justification of lines 1–3 in Fig. 2

Line 1 was generated without difficulty, leading to the situation in **Fig. 3**(a). When line 2 was attempted, the first line break found would have led to a line which was too tight. Fig. 3(b) shows the initial version of line 2 set with a minimum word space. This is clearly too long. If the last word is dropped and the maximum acceptable space is used, the situation in Fig. 3(c) is reached. This version of the line is too short. As a result of applying PUSHBACK, Fig. 3(d) (which is acceptable) was achieved. Line make-up proceeded without trouble until line 10, where a similar procedure was invoked, as shown in **Fig. 4**(a)–(c). The rest of the paragraph was composed without incident.

It is useful to compare these results with those of the conventional method. It is possible when using Juggle to disable recursive justification so that a line is fixed after it has once been composed. If this is done the final paragraph is that shown in **Fig. 5.** It is interesting to note that although a hyphen is inserted at the end of line 2, the next hyphen is at the end of line 13, and not line 10 as might have been supposed. This is the result of shifting part of 'question' on to line 3 which causes the rest of the text to shift up. It does, however, suggest that there is a 'problem quotient' for a particular piece of text which is independent of the method of resolution. Experience with other text seems to confirm this.

As an experiment, a substantial section of this paper was processed using the method described here. The sample contained just over 3000 words in 27 paragraphs. The recursive algorithm was invoked 28 times and was successful 19 times. Of the nine hyphens inserted seven were acceptable. When the text was processed with the recursion disabled, 20 hyphens were inserted of which 15 were acceptable and the hyphenation algorithm failed to find break points in three words.

Setting to shorter measures leads to less striking results. This is reasonable and was to be expected. Usually shorter lines contain fewer words and thus less white space. It is therefore less likely that sufficient spare space can be accumulated for inserting extra words. Similarly, removal of a word is likely to leave too much space for distribution. This is sad but not disastrous. Typically the measure chosen for a particular piece of composition reflects the quality and permanence of the final product. The obvious contrast is between newspapers and hardback books. Since short measure printing tends to be 'throwaway' printing, its quality is of lower importance. Thus a much wider range of spacing within a line is acceptable as a means of eliminating hyphens. In addition, since the object of hyphen removal is to raise the quality of the text, successful removal is less important. The effectiveness of this method seems to be roughly proportional to the measure used and it is thus most effective for those kinds of text which most merit care and attention.

A useful variant of this technique might be to combine it with a fairly sophisticated specialised hyphenator. Experiment might prove that there are classes of syllables (possibly suffixes and prefixes) which can be found and stripped with greater than average confidence. If so, then by using a hyphenator which detected these syllables together with the techniques described in this section it may be possible to provide better and more reliable hyphenation.

A final point is worthy of reflection. If you can reduce the number of hyphens needed, then you effectively reduce the failure rate of the hyphenation algorithm. A more rigorous way of expressing this is to say that the object of this exercise is the same as that of improving a hyphenator: the reduction of the number of bad line endings. It is not significant how this result is reached. This means that a technique like this may be worth considering even in circumstances when it will not give its best results, since a poor hyphenator will give less offence the less often it is called.

as far more important than symphonies. The concerto was obviously a more important form than the symphony, because it was an occasion
(a) for watching the composer himself apparently in the very act of composing.

as far more important than symphonies. The concerto was obviously a more important form than the symphony, because it was an occasion
(b) for watching the composer himself apparently in the very act of

as far more important than symphonies. The concerto was obviously a more important form than the symphony, because it was an occasion for
(c) watching the composer himself apparently in the very act of composing.

Fig. 4 Examination of the justification of lines 9–10 in Fig. 2

Our modern view of the classical era has been so much formed by the accumulated reverence for Beethoven that we accept without question the doctrine that the symphony is of all musical forms the most important, the one into which a composer must inevitably pour all his mightiest inspirations. That certainly was not the view of Beethoven's own period. The generation which had just lost Mozart and had exalted his memory to a place among the gods was inclined to regard operas and concertos as far more important than symphonies. The concerto was obviously a more important form than the symphony, because it was an occasion for watching the composer himself apparently in the very act of composing. It is difficult for us to imagine a period of musical history in which there were no classics, and in which all interest was concentrated on the newest production. The English, by commemorating Handel in Westminster Abbey in 1784, had taken the first step towards establishing a cult of 'the classics' in music, and this cult of Handel had just begun to spread from England to Germany. Johann Sebastian Bach, who for us to-day is the great classic of the eighteenth century, was practically unknown outside Leipzig. The only works of Bach which Beethoven is likely to have known were the Forty-Eight Preludes and Fugues; Forkel had just begun to awaken interest in Bach's music, and hardly any of it was accessible in print.

Fig. 5 The text of Fig. 2 set without using recursive justification

References

Chambers (1972). *Chambers Twentieth Century Dictionary*, Chambers, London.

CIT (1978). *The CIT Directory*. Cranfield Institute of Technology.

COOPER, P. I. (1967). The influence of program parameters on hyphenation frequency in a sophisticated justification program, in *Advances in Computer Typesetting: Proceedings of the 1966 International Typesetting Conference*, Institute of Printing, London.

Hart's Rules (1978). *Hart's Rules for Compositors and Readers at the University Press, Oxford*, 38th edition, Oxford University Press, London.

KNUTH, D. E. (1979). *TEX and METAFONT New Directions in Typesetting*, Part 2, Appendix H. Digital Press, Bedford, MA.

MOITRA, A., MUDUR, S. P. and NARWEKAR, A. W. (1979). Design and analyis of a hyphenation procedure, *Software—Practice and Experience*, Vol. 9 No. 4, pp. 325–337.

OCKER, W. A. (1975). A program to hyphenate English words, *IEEE Transactions on Professional Communication*, Vol. 18 No. 2, pp. 78–84.

OED (1933). *Oxford English Dictionary*. Oxford University Press, Oxford.

RICH R. P. and STONE A. G. (1965). Method for hyphenating at the end of a printed line, *Communications of the ACM*, Vol. 8 No. 7, pp. 444–445.

SISAM, K. (1929). *Word Division*, SPE Tract No. XXXIII.