Working Paper Series
ISSN 1170-487X

Semantic and generative models for lossy text compression

by Ian H. Witten, Timothy C. Bell, Alistair Moffat, Tony C. Smith & Craig G. Nevill-Manning

Working Paper 92/8
October, 1992

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Semantic and generative models for lossy text compression

Ian H. Witten

Department of Computer Science, University of Waikato, Hamilton, New Zealand
Phone (+64 7) 856–2889; fax 838–4155; email ihw@waikato.ac.NZ.

Timothy C. Bell

Department of Computer Science, University of Canterbury, Christchurch, New Zealand

Alistair Moffat

Department of Computer Science, University of Melbourne, Melbourne, Australia

Craig G. Nevill-Manning

Department of Computer Science, University of Waikato, Hamilton, New Zealand

Tony C. Smith

Department of Computer Science, University of Calgary, Calgary, Canada

Abstract

The apparent divergence between the research paradigms of text and image compression has led us to consider the potential for applying methods developed for one domain to the other. This paper examines the idea of "lossy" text compression, which transmits an approximation to the input text rather than the text itself. In image coding, lossy techniques have proven to yield compression factors that are vastly superior to those of the best lossless schemes, and we show that this is also the case for text. Two different methods are described here, one inspired by the use of fractals in image compression. They can be combined into an extremely effective technique that provides much better compression than the present state of the art and yet preserves a reasonable degree of match between the original and received text. The major challenge for lossy text compression is identified as the reliable evaluation of the quality of this match.

Introduction

In attending recent Data Compression Conferences we have been struck by the apparent divergence between the research paradigms of text and image compression [1, 2]. Schemes for text compression are invariably reversible or "lossless," whereas although there certainly exist lossless methods of image compression, by far the majority of contemporary research seems to be on irreversible or "lossy" techniques such as transform coding, vector quantization, and fractal approximation. This divergence is unfortunate because the opportunity for symbiosis between the two regimes is lost, and advances in one domain have negligible impact on the other. Although there are superficial reasons why one might choose to neglect the topic of lossy text compression—such as the difficulty of evaluating the quality of the resulting transmission—we have, on reflection, been led to believe that a great deal can be gained by taking seriously the idea of approximate compression of text.

Everyday experience abounds with examples of approximate compression. The art of précis, for example, is lossy compression *par excellence* and is widely used for a variety of practical purposes, though—so far—in manual rather than automatic implementations. Further examples, at a much higher compression rate, occur in newspaper headlines, the creation of which is an art that blends current affairs with an almost poetic feeling for words and their juxtaposition. Finally at the extreme end of the scale is the trash can, surely the epitome of irreversible compression.

Only the last of these three scenarios has been automated, and then only fairly recently (although the UNIX `/dev/null` is an earlier, functionally similar, implementation). The only other lossy technique of which we are aware is the commonly-suggested device of omitting vowels from text, which sacrifices readability for compression and is hardly suitable for practical use. It seems likely that every letter-based approach to lossy compression is doomed to suffer this disadvantage, and this observation led us to base our work on units larger than letters—a tactic that has precedent in lossless compression too [3, 4].

This paper describes two novel techniques for lossy compression of text. The next section develops a semantic approach that uses an auxiliary thesaurus. However, its word-by-word approach fundamentally limits the amount of compression that can be achieved. This led us to consider syntactic techniques—which we had initially discarded as apparently less powerful—for the *generation* of approximate text, and the resulting scheme, inspired by recent work on fractal compression [5], is described next. As when fractals are applied to image coding, astonishing compression factors can be achieved for

certain data sets, but the process is exceedingly time-consuming and it is not clear whether the method can be extended to apply to all texts. The final section points the way to a synthesis of the semantic and generative approaches, which promises to form an extremely powerful general method of lossy text compression.

Word-by-word semantic compression

Excellent, comprehensive thesauri have recently become available in machine-readable form (e.g. [6]) and already some compression researchers have begun to take advantage of them (e.g. [7]). This immediately suggests a macro word-replacement strategy for lossy compression: replace each word in the text with a shorter equivalent from a thesaurus. Despite its remarkable simplicity, this technique provides worthwhile compression with little semantic loss—sometimes the richness and literary texture of the prose even improves. Here is an example of the first paragraph of this paper, so compressed:

> In waiting new Data Compression Chats we let been struck by the true divergence mid the dig plans of work and bust compression [1, 2]. Cons for book compression are invariably reversible or "lossless," as as there be lossless uses of icon compression by far the top of new dig looks to be at irreversible or "lossy" ways like as vary lawing, vector quantization, and fractal copy. This divergence is sad due to the gap for symbiosis mid the two rules is past, and goes in a sod own off box at the more. As there are crude sees why a pep opt to omit the item of lossy work compression—such as the jam of trying the top of the ending transmission—we buy, on dig, been led to buy that a fat buy WC be wined by taking sadly the yen of put compression of item.

This reduces the example paragraph from 12 lines to only 9, giving a compression figure of 75%.[1] Furthermore, some common operations, such as word-counting, still function correctly on the compressed text.

A striking advantage of this new technique over virtually all known lossless methods for compression (though a notable exception appears in [8]) is that it can be re-applied to the same text with a further gain in compression performance. Naturally the semantic loss tends to increase every time the transformation is reapplied—this is an inevitable price of improved compression. For example, a second iteration of the method on the first paragraph of the paper yields:

> In waiting odd Data Compression Raps we let been struck by the due divergence mid the cut maps of go and dud compression [1, 2]. Fobs for get compression are invariably reversible or "lossless," as as there be lossless uses of god compression by far the key of odd cut sees to be by irreversible or "lossy" uses akin as go acting, vector quantization, and fractal act. This divergence is sad due to the gap for symbiosis mid the two plan is old, and goes in a sod own

[1]Though we have found that these measures do depend somewhat on the formatting of the text.

> lax jar at the over. As there are raw sees why a go opt to cut the part of lossy go compression—such as the ram of hard the key of the ending transmission—we buy, by cut, been led to win that a fat get WC be wined by robing sadly the yen of air compression of text.

The reduction is a further line, or 11.1%.

Repeated applications tend to converge rather quickly to a version that we call an *attractor* of the original paragraph (following the terminology adopted in non-linear dynamics [9]). An attractor of the example paragraph is reached after a further 6 iterations, and is not markedly different from the second-iteration version above. Extensive tests have shown that the average distance to an attractor is about 7.28 iterations, although this does vary with the style of text. Different replacement strategies can yield different attractors, and we define the *attractor set* of a given text in the obvious way. Analysis of a large number of attractor sets shows that, as one might expect, the members of the set generally bear a strong resemblance to each other, giving a small but useful degree of variation in the compressed text.

We have experimented with improved methods of word-by-word semantic compression. The basic idea is to generalize to an *expanded attractor set* by progressing *up* the semantic hierarchy before each iteration. This tends to produce slightly better compression at the expense of semantic accuracy. Unfortunately the improvement is not guaranteed, for we have discovered texts on which the generalization produces worse compression than any member of the original attractor set.

A possible solution to the problem involves a kind of simulated annealing process, which is controlled by a monotonically decreasing temperature variable. The procedure replaces a word by one at a level above in the hierarchy with a probability that depends on the current temperature value. This probability steadily approaches zero as time progresses. The operation proceeds in cycles: in every cycle each word has an opportunity to move up the hierarchy before being replaced by a shorter equivalent, and at the end of the cycle the temperature decreases according to a predetermined schedule. We are presently working on a probabilistic convergence theorem for the scheme.

All these word-by-word semantic schemes suffer a common flaw: they can never reduce the number of words in the text. Examination of human-generated précis reveals that they do not suffer the same disadvantage. After briefly considering, and discarding as insufficiently powerful, schemes that locate phrases that occur in a thesaurus and replace them by a single word, we discovered, almost serendipitously, an entirely different approach.

Generative compression of text in the style of Hemingway

The final abstract expression of every art is a number—Wassily Kandinsky, 1912

Parallel work on story generation [10] led us to consider the question of generating works in a particular literary style. Hemingway's prose was selected as a strongly characteristic and easily identified style, of which many genuine examples are available, along with some notable imitations [11]. Linguistic analysis of this corpus in terms of both the syntactic constructions commonly adopted, and the semantic entities that form the centerpiece of many Hemingway stories, resulted in a program that generates appropriately styled "pseudo-texts." Here is a brief example of output from the generating program:

> The old man who had waited for the old beggar from Madrid was certain that the locals had argued with his martini and should have argued with the waiter. Only he had not sat beside the waiter. No one but he had tried to fool the old beggar from Madrid he had heard about and knew that the bullfighter had not argued with the waiter. Only he knew that the parrot had told him about the matador's friend on Kilimanjaro. The old man knew that the locals who had not sat beside the American girl had argued with the old beggar from Madrid in a well lighted room and believed that the locals had joined up with his martini in the café. The old man had argued with his martini while fast asleep and believed that she who had not joined up with the American girl had not waited for the American girl. The old man had not cheated the matador's friend with a certain understanding. The old man was certain that the small dog with three legs who should have joined up with his martini had not waited for the old beggar from Madrid and had not cheated death. The old man had not sat beside death at the corner table.

(Note: capitalization and formatting were added to this story by hand, though we are confident that we will be able to generate these features automatically very shortly.)

Although we have been experimenting with this program for some time, only recently have we come to realize its potential import for compression. The individuality and variety in the stories is attributable to the use of a random number generator, which produces a sequence that depends solely on an initial "seed." For example, from a different seed we are able to grow the following story that begins in the same way:

> The old man had waited for his martini. The old man had not tried to fool death he had heard about. He who knew that the small dog with three legs had told him about the matador's friend felt that the bullfighter had not told him about the old beggar from Madrid while fast asleep. He felt that the locals who had not seen the old beggar from Madrid had not waited for death for nothing. The old man thought that the man with the patch over one eye had brought him the waiter in the café. The old man who should have tried to fool the waiter had not brought him the waiter and knew that the bullfighter who had not brought him his martini had not told him about death he had heard about. The old man had not sat beside death at the corner table. No one but he who had sat beside the American girl believed that the bullfighter should have brought him the waiter. He who knew that she had joined up with his martini for

> nothing was certain that the bullfighter should have tried to fool the matador's friend in a well lighted room and had not cheated the old beggar from Madrid while fast asleep.

Analysis of the algorithm reveals that the seed has just 2^{32} possible values and can therefore be stored in 32 bits [12]. This immediately suggests representing a story by its seed, thereby achieving very substantial compression of a magnitude that has never previously been realized in text compression. Since these stories average a little over one thousand characters, each normally requiring one byte, the compression factor of this technique is around 250:1. We are working on extending the program to generate longer stories and thus reap even greater compression gains.

This technique produces lossless codes for a particular class of texts: namely, those generated by the Hemingway pseudo-text program. A crucial insight is that with absolutely no modification it can actually produce *lossy* codes for a much larger class of texts. Of course, worthwhile compression with reasonable fidelity can only be expected on stories within the Hemingway genre on which the program is modeled; nevertheless this does comprise a substantial number of samples. We are working on the creation of lossy codes for all the short stories in the collection *The Snows of Kilimanjaro*. Presently the matching operation is done manually, which is a rather tedious process: we have plans to automate it as the next stage of the project.

Synthesizing the semantic and generative approaches

"The alert reader will no doubt have anticipated the next step: to combine the semantic and generative approaches to provide a more powerful approximate compression technique." Although with hindsight this seems to follow logically enough from the foregoing presentation, it nevertheless eluded us for some considerable time, and—astonishing as it might sound—was actually suggested *by the program itself*! As we pored over yet another tale of the old man, the American girl and the matador's friend late one night we were first puzzled, then incredulous, to read the very sentence that begins this paragraph, made as an aside to the waiter as the old man sat contemplating death at a corner table in a well lighted room. This may well be the first instance of a program literally suggesting an enhancement of itself to its creators. We quaffed our martinis and immediately set to work.

The semantic and generative approaches can be combined in two distinct ways. The thesaurus can be used to increase the match between a generated story and the one to be compressed; we call this "semantic enhancement." Or it can be used to decrease the size of the generated story through the normal semantic compression procedure: this is "lexical contraction." Although lexical contraction does not reduce the actual bit rate,

since the story is already represented as only four bytes, controlled experiments with human subjects, who had already been exposed to our earlier compression technique, showed that it increases the verisimilitude of the compressed text—the resulting taut, brusque prose accords better with the reader's idea of how a compressed version should read than the original, more florid, language. A lexical contraction of the first example pseudo-text above is:

> The old guy who had bided for the old bum from Madrid was set that the folks had rowed mid his martini and must get bugged too the waiter. One he had not sat at the waiter. No a yet he had sure to ass the old bum from Madrid he had heard re and knew that the bullfighter had not irked mid the waiter. Odd he knew that the parrot had told him re the matador's pal by Kilimanjaro. The old rig knew that the folks who had not sat on the yank kid had rowed mid the old bum from Madrid in a far lit live and bought that the folks had wedded up and his martini in the bar. The old man had irked and his martini as lax idle and bought that she who had not wedded up and the yank kid had not waited for the yank kid. The old arm had not conned the matador's pal mid a set wit. The old arm was set that the off pup mid three arms who must use wedded up mid his martini had not held for the old bum from Madrid and had not fobbed ruin. The old guy had not sat on ruin on the jam list.

Semantic enhancement is clearly the more powerful combination. Compared with the rather stilted vocabulary of the raw pseudo-text, semantic substitution offers much richer and more variegated language. For instance, here is a transformation of the first example pseudo-text above:

> The perennial gear who had procrastinated for the archaic beggar from Madrid was unquestionable that the near-at-hands had battled additionally his martini and must concede haggled among the waiter. Peerless he had not sat around the waiter. No one though he had infallible to inveigle the obsolete drifter from Madrid he had learned about and knew that the bullfighter had not warranted midst the waiter. Solely he knew that the parrot had told him respecting the matador's promoter atop Kilimanjaro. The perennial homo sapiens knew that the folks who had not sat around the yankee adolescent had scrapped in addition the ancient mendicant from Madrid in a ruddy delicated elbowroom and knowed that the verging ons had fused jack up midst his martini in the tearoom. The dead male had irked with his martini whereas precipitous motionless and gathered that she who had not tied boost within the yankee coed had not delayed for the American daughter. The past fortify had not bilked the matador's companion within a factual insight. The outmoded widower was stated that the limited pup moreover three legs who should have laced acquainted inside his martini had not waited for the grizzled pauper from Madrid and had not robbed passing. The passé fellow had not sat nearby decease atop the bottle up remit.

The much larger space of possible compressed texts that can be created with this method does exacerbate the problem, mentioned above, of finding the best match to a given source text. Efficient algorithms for this task are the subject of a forthcoming publication.

It may not be apparent how a text that has been generated and subjected to semantic enhancement can be coded efficiently. Although four bytes suffice to represent the

original pseudo-text, it seems to be necessary to specify the enhancement individually for each word, thus negating the tremendous compression that the generative method yields. Fortunately, the problem can be solved very simply. Careful examination of the program reveals that sentence enhancement, like story generation, is fully characterized by a 32-bit random number generator seed. This seed is all that is needed to regenerate the enhanced text without any loss of fidelity. Thus a total of 8 bytes is necessary for lossy compression of a text of any size: 4 for the generator and 4 for the semantic enhancement. We believe that still further gains may be had by deriving one of the seeds from the other via an appropriately parameterized transformation; this is the subject of ongoing research. However, 8 bytes is already a rather efficient representation and the potential for improvement is small, perhaps insignificant.

Conclusions

Tall oaks from little seeds grow—David Everett 1769-1813 (adapted)

This paper has illustrated the benefits that can be reaped by taking the idea of lossy text compression seriously and adapting some of the techniques from the image compression world. Thesaurus substitution is a straightforward technique that results in appreciable compression: it has the advantage that, up to a point, it can be applied repeatedly to further reduce the size of the compressed text. However, it suffers from the serious disadvantage that although it reduces the size of each individual word, it can never reduce the *number* of words in the text.

We were thus led to consider generative techniques and the coding of a story in terms of the random number seed from which it grew. This technique, which gives remarkably effective lossless compression for a restricted class of texts, can be viewed as a lossy compression method for a more general class—indeed, for a complete genre. The verisimilitude of the compressed stories can be increased by a further phase of thesaurus substitution, to ensure that the reader perceives them as compressed. Alternatively, their accuracy can be increased through what we have called "semantic enhancement." Although this technique doubles the bit rate of the compressed text from 4 to 8 bytes, it permits a much more accurate rendering of the original text. One criticism of the scheme is the slow encoding speed; however, this is more than made up for by very fast decoding.

At present we are investigating two other methods of lossy text compression. The first uses *character* substitution rather than *word* substitution. For example, letters such as "q" and "x" can be eliminated by substituting phonetically equivalent letters, yielding phrases such as "kwik brown foks". This can be represented in fewer bits, yet conveys the same

information (when read aloud, at least). The second technique is based on progressive image transmission [13]. For example, in progressive text transmission of a paper we first send the title, then section headings, the abstract, the conclusion, and so on. The more of this representation that is stored or transmitted, the more lossless the representation is. In experiments with transmitting papers, it appears that most of the time users cancel transmission quite early on, achieving significant savings in transmission costs (although this could be because we are using our own papers in the trials).

Undoubtedly the largest problem for lossy text compression is the question of evaluating the texts produced, and providing satisfactory measures of their subjective "quality." This is a problem that has also exercised the image coding community over the years and will undoubtedly succumb to the same sort of solution; namely to adopt a standard story, perhaps a biography of the infamous "Lena" [14], as the subject of *all* compression experiments so that they can be compared on the same basis.[2]

Acknowledgment

We thank the DCC'93 organizers for (perhaps unwittingly) stimulating us to write this paper by scheduling the conference to include April 1.

References

[1] Storer, J.A. and Reif, J.H. (Editors) (1991) *Proceedings Data Compression Conference*. IEEE Computer Society Press, Los Alamitos, CA.

[2] Storer, J.A. and Cohn, M. (Editors) (1992) *Proceedings Data Compression Conference*. IEEE Computer Society Press, Los Alamitos, CA.

[3] Moffat, A. (1989) "Word-based text compression," *Software—Practice and Experience* 19(2): pp. 185–198; February.

[4] Horspool, R.N. and Cormack, G.V. (1992) "Constructing word-based text compression algorithms," in [2], pp. 62–81.

[5] Barnsley, M.F. and Sloan, A.D. (1988) "A better way to compress images," *Byte*, January 1988, pp. 215–223.

---

[2] A brief note for the curious. "Lena" or "Lenna" is a digitized Buck centerfold. Lena Soderberg (née Sjooblom) was be popped keep in her folk Sweden, well married and three boys and a art and the air grog lock. In 1988, she was seen by a Swedish computer akin tome, and she was fairly charmed by what had done to her art. That was the top she knew of the way of that oil in the computer job. The item in the January 1992 end of *Optical Engineering* (v. 31 no. 1) data how Buck has lastly caught at to the life that their claim on Lenna Sjooblom's slide is man bigly defied. It arms as if you wish get to nab grant from Stud to blaze it in the next.

[6] Roget, P.M. (1911 edition) *Roget's thesaurus of English words and phrases.* Available from Project Gutenberg, Illinois Benedictine College, Lisle, IL (ftp mrcnext.cso.uiuc.edu).

[7] Nevill, C. and Bell, T. (1992) "Compression of parallel texts," *Information Processing and Management* 28(4).

[8] Schnapp, R. (1992) "Instant Gigabytes?" *Byte* 17(6): p. 45; June.

[9] Gleick, J. (1988) *Chaos: making a new science.* Heinemann, London.

[10] Smith, T.C. and Witten, I.H. (1991) "A planning mechanism for generating story text," *Literary and Linguistic Computing* 6(2): pp. 119–126.

[11] Plimpton, G. (1989) *The best of bad Hemingway.* Harcourt Brace Jovanovich, New York.

[12] *Unix programmer's manual.* (1984) 4.2 Berkeley Software Distribution. Chapter 3C: RAND.

[13] Tzou, K.-H. (1987), *Progressive image transmission: a review and comparison of techniques.*, Optical Engineering 26(7): pp. 581–589.

[14] Sjooblom, L. (1972) *Playboy*, p. center; November.