Semantic and generative models
for lossy text compression

Ian H. Witten
Department of Computer Science, University of Waikato, Hamilton, New Zealand.
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.
Harold Thimbleby
Department of Psychology, University of Stirling, Stirling, Scotland.[†]

Abstract
The complementary paradigms of text compression and image compression suggest that there may be potential for applying methods developed for one domain to the other. In image coding, lossy techniques 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. This paper investigates the resulting tradeoff between subjective quality of the transmission and its compression factor. Two different methods are described, which can be combined into an extremely effective technique that provides far better compression than the present state of the art and yet preserves a reasonable degree of perceived match between the original and received text. The major challenge for lossy text compression is the quantitative evaluation of the quality of this match.

Introduction
We have been struck by the apparent divergence between the research paradigms of text and image compression [1, 2], despite the fact that both are concerned with compressing information whose subjective quality must be recoverable. Schemes for text compression are invariably reversible or lossless, whereas although there certainly exist lossless methods of image compression, much research effort addresses irreversible or lossy techniques such as transform coding, vector quantization, and fractal approximation.

The divergence between the text and image paradigms is unfortunate because the opportunity for symbiosis between the two approaches 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 regenerated message — in this paper we suggest that a great deal can be gained by taking seriously the idea of approximate compression of text.

CONTRIBUTIONS AND STRUCTURE OF THE PAPER
We have developed two novel techniques for lossy compression of text, and they are described in sufficient detail for our work to be replicated and tested in realistic compression situations. The necessarily very short texts that are used in the examples exhibited in this paper, along with their statistical limitations, can be no more than indicative of the underlying power of the techniques we describe.

[†]Address for correspondence. Phone (+44) 786 467679; fax 786 467641; hwt@compsci.stirling.ac.uk.

The next section develops a semantic approach that uses an auxiliary thesaurus. However, any word-based approach limits the amount of compression that can be achieved if used in isolation. Hence we next consider syntactic techniques, using fractal compression [3], for the *generation* of approximate text. As when fractals are applied to image coding, extremely high compression factors can be achieved for certain data sets, but the process is time-consuming and it is not clear whether the method can be extended to apply efficiently to all texts.

The final section points the way to a synthesis of the two approaches, which together promise to form an extremely powerful and yet general method of lossy text compression.

Lossy text compression: background and motivation

Everyday experience abounds with examples of approximate text compression. The art of précis, for example, is lossy compression *par excellence* and is widely used for a variety of practical purposes, though 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! While the last example may seem flippant (though it is distinguished as the only one that has been machine-implemented to date), the point is that lossy compression can serve a wide range of purposes. The examples also show that — as in image compression — perceived quality is not easily specified.

In synthetic languages the lossy text compression problem is easy to specify: compress the text (making suitable lexical and syntactic transformations) but preserve semantics. In a programming language such as Pascal, simple lossy compression may be achieved by removing superfluous white space. This yields substantial compression using a trivial algorithm. Literate programming's notion of 'tangling' is an example of lossy compression that not only loses layout information but all comments as well [4]. Further compression can be achieved by applying compiler optimisation methods to the text (optimising for source code length, rather than object code length), and for example, may result in variable names being shortened or even lost, or expressions being rewritten.

In natural language, other lossy techniques include the commonly-suggested device of omitting vowels from text. This sacrifices readability for compression and is hardly suitable for practical use, though variations are used in shorthand (both speed writing and non-roman scripts), Braille and stenography. Thus, Dearborn's Speedwriting (1924) was designed for typewriter use, using standard letters and punctuation. For example, the code eC represents the sound *each*, the C designating the sound *ch*. Sixty rules in Speedwriting provide for lossy compression of a vocabulary of around 20,000 words. The Soundex system uses a simpler lossy compression technique to avoid problems of requiring exact spelling matches for text database searches. (The methods described in this paper share this useful property.)

At first sight the historical standing and simplicity of sound-based lossy compression might seem very attractive. Its effectiveness is easily demonstrated. The optimal encoding of "quick brown fox" using an order–0 model of English (Brown Corpus) is 81.3 bits; after making the lossy phonetic transcription x cs, qu cw, k c (obtaining "cwic brown focs," which sounds the same when read aloud) it compresses to 68.6 bits using the appropriately modified order–0 model. Surely a dictionary-based phonetic model would do better? In fact a conventional order–n model can do even better, since it necessarily models text that is pronounced: it is effectively a phonetic model for sounds of at most $n+1$ characters. Not only can an order–n model compress more effectively, but it can do so losslessly!

The compression of the various shorthand techniques results from the poor correspondence of single letters to sounds; moreover, every letter- or phoneme-based approach to lossy text compression assumes an appropriate phonetic model to intepret it. Conventional lossless compression can be far more effective, in terms of compression, but obviously results in unreadable data that requires a computer to unencode. Such observations lead us to base the text compression work to be described on *semantic* units

larger than letters (and independent of sound), a tactic corresponding to one that has precedent in lossless compression [5, 6].

Word-by-word semantic compression

Excellent, comprehensive thesauri have recently become available in machine-readable form (e.g., [7]) and already some compression researchers have begun to take advantage of them (e.g., [8]). Thesaurus compression is a macro word-replacement strategy for lossy compression: replace each word in the text with a shorter equivalent form taken from a (given) thesaurus. Despite its remarkable simplicity, this technique provides worthwhile compression with little semantic loss — sometimes the richness and literary texture of the prose improves.

Here is an example of the first two paragraphs of this paper, so compressed:

> We win been struck by the true divergence mid the dig plans of book and bust compression [1, 2], dig the life that both are scary and jamming lore whose biased top must be recoverable. Cons for tome compression are invariably reversible or lossless, as as there well be lossless uses of idol compression, ton dig go owners irreversible or lossy ways like as vary lawing, vector quantization, and fractal guess.
>
> The divergence mid the work and bust plans is sad due to the gap for symbiosis mid the two approaches is gone, and goes in a area win off tap at the some. As there are brief wits why a pep opt to omit the item of lossy work compression — like as the fix of trying the top of the regenerated sense — in this page we jog that a key buy wc be got by asking sadly the yen of put compression of tome.

This reduces the text from 1031 bytes to only 804, giving a compression figure of 78%. Of course, the word count is unaffected and therefore common operations, including word-counting, still function correctly on the compressed text.

A striking advantage of the thesaurus technique, particularly in comparison with lossless methods for compression (though a notable exception appears in [9]) is that it can be re-applied to the same text with a further gain in compression performance. The semantic loss tends to increase every time the transformation is reapplied — this is the inevitable price of improved compression. For example, a second iteration of the method on the first two paragraphs of the paper yields:

> We buy been struck by the due divergence mid the cut maps of bag and dud compression [1, 2], dig the root that both are dire and baring data whose biased bow must be recoverable. Lies for opus compression are invariably reversible or lossless, as as there fit be lossless uses of god compression, ton cut go heads irreversible or lossy ways will as go lawing, vector quantization, and fractal go.
>
> The divergence mid the do and bag aims is sad due to the gap for symbiosis mid the two approaches is lost, and goes in a sod net lax dab on the a. As there are tell gags why a go opt to bar the text of lossy do compression — such as the fit of hard the key of the regenerated wit — in this hail we jog that a tip win wc be won by begging sadly the yen of put compression of text.

The reduction is a further 3.7%.

Repeated applications tend to converge rather quickly to a fixed point that we call an *attractor* of the original paragraph (following the terminology of non-linear dynamics [10]). An attractor of the example paragraph is reached after a further 6 iterations, and hardly differs from the second-iteration version above, except under very careful examination — in other words, its deep meaning is preserved.

Tests show that the average distance to an attractor is about 7.28 iterations, although this varies with the style of text and the particular thesaurus. Different replacement strategies can yield different attractors; 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 can construct texts on which the generalization produces worse compression than any member of the original attractor set.

A possible solution to this degeneration involves a simulated annealing process. 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 the predetermined schedule. (We are working on a probabilistic convergence theorem for the scheme, and will report on it in due course.)

All word-by-word compression schemes suffer a common flaw: they can never reduce the number of words in the text. (Human-implemented lossy compression, such as précis, does not necessarily suffer the same disadvantage.) We discarded as insufficiently powerful converse schemes that locate phrases that occur in a thesaurus and replace them by a single word. Another method is required; we turn to one possibility in the next section.

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 [11] led us to consider the question of generating works in a particular *literary style*. Hemingway's prose was selected arbitrarily to illustrate the method; it has a strongly characteristic and easily-identified style (many genuine examples are available, along with some notable imitations [12]). Linguistic analysis of the available 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 generated texts in the chosen genre. Here is a brief example of output:

> 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 Madridand had not cheated death. The old man had not sat beside death at the corner table.

The individuality and variety in the stories is directly attributable to the use of a pseudo-random number generator, which produces a sequence that depends solely on an initial seed. From a different seed one may grow a different text, within the constraints of the genre. For example, here is a text 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.

The number generator has one seed, hence just 2^{32} possible states [13]. Consequently any text so generated can therefore be stored in 32 bits. The seed represents a very substantial compression (indeed, of a magnitude that has never previously been realized in text compression).

This technique produces lossless codes for a *particular* class of texts: namely, those generated by the Hemingway pseudo-text program.

The crucial insight is that with no modification it can 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, in our experiments 2^{32}. 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 next step is to combine the semantic and generative approaches to provide a more powerful approximate compression technique.

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 bit rate (the story is already represented by 32 bits), 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 generated 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 compression 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 one such transformation of the first sample text:

> 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.

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 compression that the generative method yields. Fortunately, the problem can be solved very simply.

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. Still further gains may be had by deriving one of the seeds from the other via an appropriately parameterized transformation. However, 8 bytes is already a rather efficient representation and the potential for further improvement is small, perhaps insignificant.

Conclusions

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

This paper has illustrated the benefits that are obtained by taking the idea of lossy text compression (itself motivated by lossy image 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. Generative techniques and the coding of a text in terms of a random number seed gives remarkably effective lossless compression for a restricted class of texts, and can be viewed as a lossy compression method for a more general class — indeed, for a genre. The verisimilitude of the compressed stories can be increased by thesaurus substitution, to ensure that the reader perceives the result as compressed. Alternatively, accuracy can be increased through semantic enhancement. Although this technique doubles the bit rate of the compressed text, 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 the very fast decoding that is possible.

We are also investigating other methods of lossy text compression. One technique that shows promise is based on progressive image transmission [14]. 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 (for assessing images) and will undoubtedly succumb to the same sorts of solution; namely, in the text case, to adopt a standard text, perhaps a biography of the infamous Lena [15],

as the subject of *all* compression experiments so that they can be compared on the same benchmark.[1]

Acknowledgment

There is no-one to thank. We bear full responsibility.

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

[4]   Knuth, D.E. (1984) Literate Programming, *Computer Journal,* **27**(2): pp. 97–111.

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

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

[7]   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).

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

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

[10] Gleick, J. (1988) *Chaos: Making a New Science*. Heinemann, London.

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

[12] Plimpton, G. (1989) *The Best of Bad Hemingway*. Harcourt Brace Jovanovich, New York.

[13] *Unix programmers manual*. (1984) 4.2 Berkeley Software Distribution. Chapter 3C: RAND.

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

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

---

[1]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 Sjoobloms slide is man bigly defied. It arms as if you wish get to nab grant from Stud to blaze it in the next.