A-022

可逆型情報秘匿を利用した符号化データの再圧縮方法
Recompression method of coded data using reversible information concealment

岡崎　優佑 †　　伊藤　浩 †
Yusuke Okazaki　　Hiroshi Ito

1. はじめに

近年、ハフマン符号、ランレングス符号など様々な圧縮技術が生まれ、活用されてきた。しかし、これらの可逆型圧縮技術には冗長性があり、圧縮効率を向上させる余地は残っているといえる。また、暗号化データに情報を付与するものや、符号化データに情報を付与するものは多い [1] が、暗号化と符号化の両方を施したデータに情報を付与できるものは少ない。

本文では、圧縮データに対して、そのデータの一部を別の部分に秘匿することによりデータを再圧縮する方法を提案する。圧縮データは暗号化されていても良い。秘匿は埋め込む情報により異なる箇所を反転させることで行い、反転箇所を原データの規則性を用いて特定することで情報の伝達を可能にする。これによって秘匿した分のデータ量が減るので実質的に圧縮率を向上させることができる。

情報源には英文のテキストデータ、情報源符号化に算術符号化 [2] を用いた。規則性の基準として①文字単体の規則性、②文字間の規則性、③単語間の規則性を用い、それぞれを比較評価した。

2. 原理

図 1 は符号化から再圧縮までの流れを示す図である。図において、英文のテキストデータは算術符号で圧縮され、圧縮データは乱数との排他的論理和をとることにより暗号化される。次に、暗号化データを $N+1$bit のブロックに区切り、各ブロックの最後の 1bit をその前の Nbit に埋め込み、ブロック長を Nbit に短縮して出力する。情報を埋め込むには特定のビットを反転させる。0 を秘匿したい場合は前半のビットを反転させ、1 を秘匿したい場合は後半のビットを反転させる。ここで、秘匿する情報は暗号化データの一部であるから、この処理により秘匿した分だけ圧縮率を向上させることができる。

復号時に秘匿した情報を取り出すには以下のようにする。受信した符号データを Nbit のブロックに区切る。次に 1 つのブロックの前半のビットを反転させたデータとその区間の後半のビットを反転させたデータを作成する。それぞれのデータを復号してテキストファイルに戻すと、情報を秘匿する際と同じ箇所を反転させたファイルは元のアスキーコードに戻るため英文の一部が表示される。間違った箇所を反転させると一般に英文が復号されない。このことを利用して、英文が復号されたかどうかを見ればどこを反転させたかが分かり、情報を取り出すことができる。各ブロックにおいて、反転させたビットを元に戻し、復号した 1 ビットの情報をブロックの最後に付け加えてブロックのデータを復元する。この処理をすべてのブロックに適用すれば元の圧縮データが得られる。暗号化を行った場合は、適宜情報を平文に戻しながら上記の処理を行えば良い。

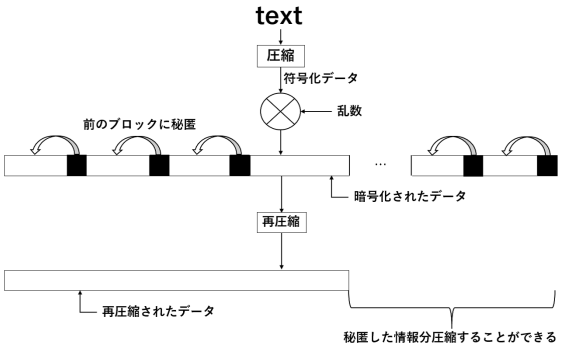

図 1: Proposed method of recompression

3. 規則性の基準

前節で述べたように正しく反転させたデータからは英文が復号される。復号されたデータが英文かどうかを判断する基準としてもっとも単純なものは、文字がアルファベットかどうかを判定することである。これを文字単体の規則性と定義する。しかし、これはしばしば不十分である。図 2 は反転後の復号文の例である。図において、下の反転が正しい文である。しかし、上の反転に対しても偶然アルファベットが復号された。このようにどちらの文もアルファベットが復号されるとアルファベットかどうかでは判別できない。

そこで第 2 の基準として文字間の規則性を用いる。これには英文を学習した RNN(Recurrent Neural Network)[3] を用いた。この RNN は入力されたテキストを文字レベルで学習し、前の文字を与えられた時に次の文字の確率分布を出力することができる。例えば、図 2 の場合、上の文と下の文で初めて異なるアルファベットが出てくるのは 26 字目の i と j である。ここで RNN の文字予測を参照すると i よりも j の方が確率が高いことが分かるため、下の反転が正しい文であると判断することができる。

しかし、この RNN は英文の意味までは理解しているとは言えない。そこで第 3 の基準として単語間の規則性を用いた。この判定は人間が行った。

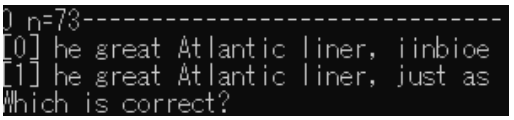

図 2: Problems in decoding

†日本大学, Nihon University

4. 遅延を用いた復号

単語間の規則性を用いたとしても確実に復号できるとは限らないので、遅延を用いた誤り訂正を行う。遅延とは、1 つのブロックを復号した時、その時点では判定を行わず、後続のブロックの復号を行い、複数ブロックの復号文から最初の 1 ブロックに秘匿された情報を特定することである。例えば、遅延長を 1 ブロックとすれば、1bit の判定に 2 ブロックを復号しているので、秘匿情報として 00, 01, 10, 11 の 4 つのパターンを検査する必要がある。00 または 01 が正しい時は 0 をそうでない時は 1 を復号する。

遅延機能は 1 度の判定で復号するデータを増加させるというメリットがあるが、1 度の判定における情報の組み合わせが指数関数的に増え、復号が複雑になるというデメリットがある。そのため、実験では 1 ブロックの遅延のみを用いた。

5. 実験

英文のテキストデータを圧縮したデータに対して前述した方法で情報を埋め込み、判定基準を変えて復号し、比較実験を行った。本来再圧縮では圧縮されたデータの一部を別の部分に埋め込むが、今回は性能の確認のため、圧縮データを Nbit 毎のブロックに区切り、各ブロックにランダムに 0 か 1 を最大 293bit 埋め込んだ。RNN の学習データにフランス人ジャーナリストの Jules Huret による舞台女優 Sarah Bernhardt の伝記の序文と 1 章（計 7536 字）を用い、再圧縮の評価にその一部の 921 字を用いる。圧縮後のデータサイズは 588byte である。規則性の基準として、①文字単体、②文字間、③単語間の規則性を用いる。

5.1 遅延なしの場合

ブロック毎に 3 つの規則性の基準のそれぞれを用いて遅延を用いずに判定を行う。N=16, 32, 48, …, 96bit の場合について、復号誤りの有無を調べた。表 1 は復号できるブロック長の比較結果である。確実に正解文を確定できる判定のみで文末まで復号できた場合を○、それ以外の場合は×とした。ブロック長の（）内の数字は埋め込みビット数である。

表 1: Results of decoding with no delay

		基準		
		①	②	③
ブロック長	16(293)	×	×	×
	32(146)	×	×	×
	48(97)	×	○	○
	64(72)	×	○	○
	80(57)	×	○	○
	96(48)	○	○	○

表 1 から遅延なしに 1 回の判定で復号できるブロック長は、①だと 96bit である。これは圧縮率に換算すると 1/96 なので 1.04%向上したことになる。②と③はどちらも 48bit なので、圧縮率に換算すると 1/48 となり、2.08%向上したことになる。

5.2 遅延復号の効果

表 2 は遅延機能を用いた場合の比較結果である。判定を行う度に遅延を 1 ブロック分だけ行った。確実に正解文を確定できる判定のみで文末まで復号できた場合を○、それ以外の場合は×とした。

表 2: Results of delayed decision decoding

		基準		
		①	②	③
ブロック長	16(293)	×	×	×
	32(146)	×	○	○
	48(97)	×	○	○
	64(72)	○	○	○
	80(57)	○	○	○
	96(48)	○	○	○

表 2 から同じブロック長に対して遅延機能を用いて復号を行うと正しく復号できる場合が増えることが分かる。1 回の判定で復号できるブロック長は、①だと 64bit である。これは圧縮率に換算すると 1/64 なので 1.56%向上したことになる。②と③はどちらも 32bit なので、圧縮率に換算すると 1/32 となり、3.13%向上したことになる。

1bit の情報を秘匿するブロック長は Nbit のまま復号時には $2N$bit の復号結果を見ることができるので、遅延機能は復号時に用いると効果的であると言える。

6. まとめ

英文の規則性を用いて圧縮データへ情報を秘匿し、圧縮効率を向上させる方法を提案した。実験の結果、遅延なしの場合は文字間の規則性と単語間の規則性を用いた場合が最も良く、圧縮率は 2.08%向上した。遅延ありの場合も文字間の規則性と単語間の規則性を用いた場合が最も良く、圧縮率は 3.13%向上した。誤り訂正として遅延を用いると、圧縮率向上に効果がある。単語間の規則性を判断基準とすることが最も良いと思われるが、人手を使わない自動的な判定方法の確立が課題である。

今回は性能確認のため秘匿情報を圧縮データの一部ではなく別に用意した。今後は、秘匿情報に圧縮データの一部を用いて検証していきたい。また、広範囲なデータで性能を確認することが必要である。誤り訂正にトレリス符号 [4] を用いれば、効率的な誤り訂正が期待できる。

参考文献

[1] W. Puech, M. Chaumont, and O. Strauss, “A Reversible Data Hiding Method for Encrypted Images,” Proc. SPIE, Vol. 6819, 2008.

[2] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic Coding For Data Compression,” Communications of the ACM, Vol. 30, No. 6, pp. 520-540, 1987.

[3] A. Karpathy, “The Unreasonable Effectiveness of Recurrent Neural Networks,” http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[4] G. Ungerboeck, “Trellis-Coded Modulation with Redundant Signal Sets,” IEEE Communications Magazine, Vol. 25, No. 2, pp. 5-21, 1987.