学 位 論 文 内 容 の 要 旨

博士の専攻分野の名称　博士（情報科学）　氏名　吉田 諭史

学 位 論 文 題 名

Efficient Variable-to-Fixed Length Coding Algorithms for Text Compression
（テキスト圧縮に対する効率よい可変長-固定長符号化アルゴリズム）

データ圧縮とは, データに含まれる冗長性を除去することで, データを格納するために必要な領域を削減する技術である. 本論文では, テキストデータに対する可逆な圧縮, すなわちテキスト圧縮について議論する. 大量のテキストデータを補助記憶装置に保存し, 後に再利用する際には, データへの I/O 速度がボトルネックとなる. もし, 圧縮データをメモリ上に展開せずに直接解析できれば, データを圧縮せずに保存した場合よりも様々な処理が高速化されるようになる. そのため, 従来の圧縮率や圧縮・展開速度などに加えて「圧縮データの取り扱いやすさ」という新しい評価基準がデータ圧縮には必要である. この基準では, 現在主流である可変長符号を利用した圧縮法は優れているとは言えない. なぜなら, 各符号語の長さが可変であるため, 符号語の境界が不明瞭で, データへの自在なアクセスが困難だからである.

「圧縮データの取り扱いやすさ」という基準からは, 固定長の符号を用いるほうが良い. この観点からは, 可変長-固定長符号 (Variable-to-Fixed Length Code), すなわち VF 符号が有望である. VF 符号は, 入力テキストを可変長の部分文字列に分解して, 各部分文字列に固定長の符号語を割り当てる圧縮法である. VF 符号は, 各符号語の境界が明確であるため, アクセス性の高いデータ圧縮の実現が期待できる. しかし, 従来の VF 符号化手法では, 可変長符号を用いる圧縮法に比べて優れた圧縮率が得られなかったため, 理論的には早い時期から研究が存在するものの, 実用的にはあまり重視されてこなかった.

そこで, 本研究では, VF 符号化手法を改善し, その圧縮率, 圧縮速度, 展開速度の三つの評価基準における性能を向上させることを目的とする. これにより, 従来の VF 符号の常識を打破し, 「圧縮データの取り扱いやすさ」を加えた四つの基準すべてにおいて, 高い水準で均整のとれたデータ圧縮法を実現することを目標とする.

初めに, 第 3 章では, Yamamoto と Yokoo らによって 2001 年に提案された AIVF 符号化の改善手法について述べる. AIVF 符号化は, 記憶のない情報源に対して, 理論的に最適な圧縮を与える符号化の一つである. 最も古典的な VF 符号である Tunstall 符号に代表される従来の VF 符号化では, 分節木とよばれる木構造を辞書として用いる. AIVF 符号は, 複数の分節木を用いることで Tunstall 符号よりも実際上優れた圧縮率を達成するが, 圧縮により多くの領域と時間が必要であることが知られている. 本章で提案する手法では, AIVF 符号の複数の分節木を一つに統合した木構造を構築し, 仮想的に元の AIVF 符号と同じ符号化を模倣する. また, 本章では, 統合された分節木のノード数と, 統合により削減されるノード数の上界と下界の理論的な解析を与える. さらに, 提案手法が, 自然言語テキスト等に対して AIVF 符号よりも高速に動作することを実験的に示す.

次に, 第 4 章では, 既存の VF 符号で構築された分節木を洗練する手法について議論する. 本章で提案する手法では, 入力テキストを繰り返し読みながら, 分節木に存在するが有用ではないノードを削除し, 分節木には存在しないが有用になると予想されるノードを追加することを繰り返す. これ

により, 最終的に高い圧縮率を達成する分節木を構築する. 実際, 喜田により 2009 年に提案された STVF 符号に対して提案手法を適用することで, VF 符号でありながら, 現在主流である Lempel-Ziv 法を土台とした可変長符号による圧縮法と同等の優れた圧縮率が得られることを示す.

次に, 第 5 章では, 文法圧縮と固定長符号を組み合わせることで VF 符号を実現する手法について議論する. 文法圧縮は, 入力テキストを一意に生成する形式文法でモデル化し, その文法を符号化することでデータ圧縮を実現する手法である. 本章では, Larsson と Moffat によって 2000 年に提案された Re-Pair アルゴリズムと固定長符号とを組み合わせた手法を与える. Re-Pair アルゴリズムは, 比較的単純な文法変換に基づく圧縮手法であるが, 非常に良好な圧縮率を達成する手法である. 提案手法では, 辞書と圧縮データの量の総和を評価する計算式を導入し, Re-Pair アルゴリズムによって得られた文法から辞書に含める文法規則を取捨選択する. 分節木を用いる従来の VF 符号の枠組みを超えた本手法により, 前章で提案した手法を上回る圧縮率を達成すると同時に, 上述した STVF 符号などに比べて数十倍高速な圧縮速度を実現することができる. また, 展開速度に関しても, 既存のデータ圧縮法の中で最高水準の速度を実現する.

第 6 章では, 前章までで述べた VF 符号化を大規模なテキストに対して適用する際に必要となる応用技術について議論する. 具体的には, 大規模テキストのブロック化と辞書共有に基づく圧縮手法と, 元テキスト上での指定された位置に対応する圧縮データ上の場所に高速にアクセスする手法の二つである. 前者は, 圧縮対象のテキストを固定長のブロックに分けて圧縮することで, 使用するメモリ量を削減する手法である. その際, ブロック間で生成される辞書を共有することで圧縮率の向上を図る. 後者について, 圧縮データ上の対応する位置を特定するためには, 通常, 圧縮テキストを最初からその位置まで展開しなければならない. これに対し, 提案手法では, n ビットのビット列とその簡潔な完備辞書を追加することで, この問題を高速に解くことができる. ここで, n は圧縮対象のテキストの長さである. また, 提案手法が, Maruyama らによって 2013 年に提案された, 文法圧縮に基づく可変長符号化方式である FOLCA よりも, 十倍程度高速にデータアクセスできることを実験的に示す.

以上をまとめると, 本論文では, 圧縮率, 圧縮速度, 展開速度の三つの評価基準において, 可変長符号を用いた既存の圧縮法に比肩する性能を達成する VF 符号化手法を実現することができた. これにより, 「圧縮データの取り扱いやすさ」を加えた四つの基準すべてにおいて高い水準の性能を有するデータ圧縮法を実現した.