第156回 データベースシステム研究会,
2012.12.12, 京都

共有辞書を用いた効率の良い圧縮アルゴリズム

Variable-to-Fixed-Length Encoding for Large Texts Using a Re-Pair Algorithm with Shared Dictionaries

◎ 関根 渓[†], 笹川 裕人[†], 吉田 諭史[†], 喜田 拓也[†]

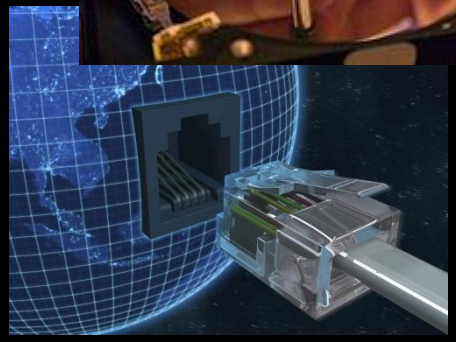
[†]北海道大学大学院情報科学研究科

背景：巨大なデータ

- 計算機上で扱うデータの巨大化

容量が足りない

時間がかかる



- データ圧縮は非常に重要

背景：文法圧縮

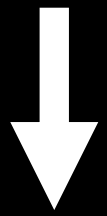
- 近年，文法圧縮に注目が集まっている.
  - データを一意に生成する文法を構築し，その文法を符号化する圧縮手法.
  - データの文脈を捉えられる.

- 代表的な文法圧縮アルゴリズム
  - Re-Pair [Larsson and Moffat 1999]
  - Re-Pair-VF [Yoshida and Kida 2012]
  - SEQUITUR [Nevill-Manning et al. 1994]
  - BPE [Gage 1994]

Re-Pair-VF アルゴリズム

- 最頻出の2-gramを新しい記号で置き換えていく.

ENOOOBOEOOOBEEOOOB



ENFOBOEFOBEEFOB

F → OO

Dictionary

Re-Pair-VF アルゴリズム

- 全ての2-gramがユニークになったら変換終了

T: ENOOOBOEOOOBEEEOOOB

ENFOBOEFOBEEFOB

ENABOEABEEAB

ENCOECEEC

ENCODED

圧縮

伸展

Dictionary

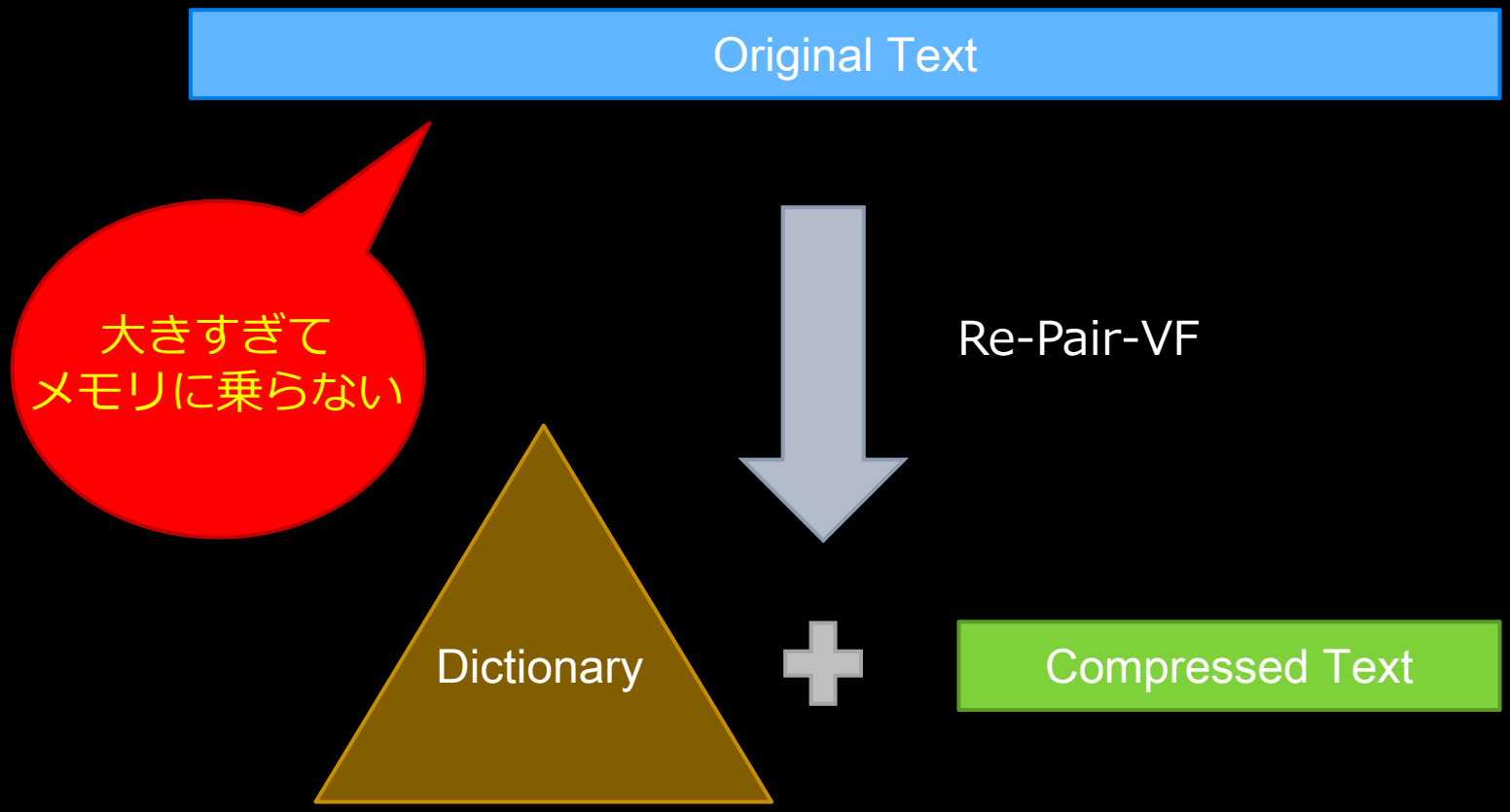
F → OO
A → FO
C → AB
D → EC

固定長符号で符号化

Re-Pair-VF アルゴリズム

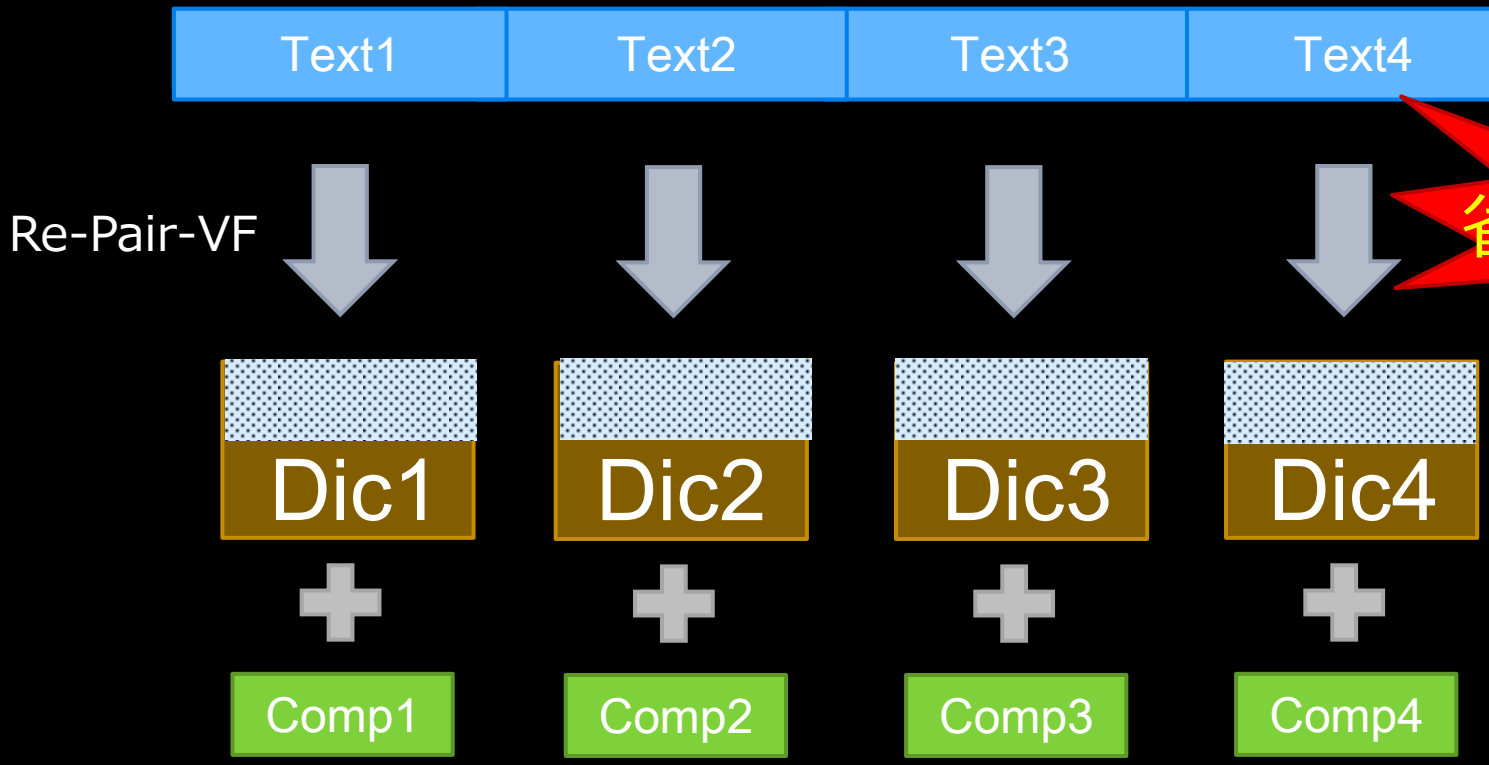
- 利点
  - 圧縮率がよい. （gzipを超える）
  - 圧縮テキストが扱いやすい.
    （圧縮パターン照合が高速に出来る）
- 欠点
  - メモリ消費が激しい. （入力テキストの10~20倍）

Re-Pair-VF アルゴリズム



対策：テキストのブロック化

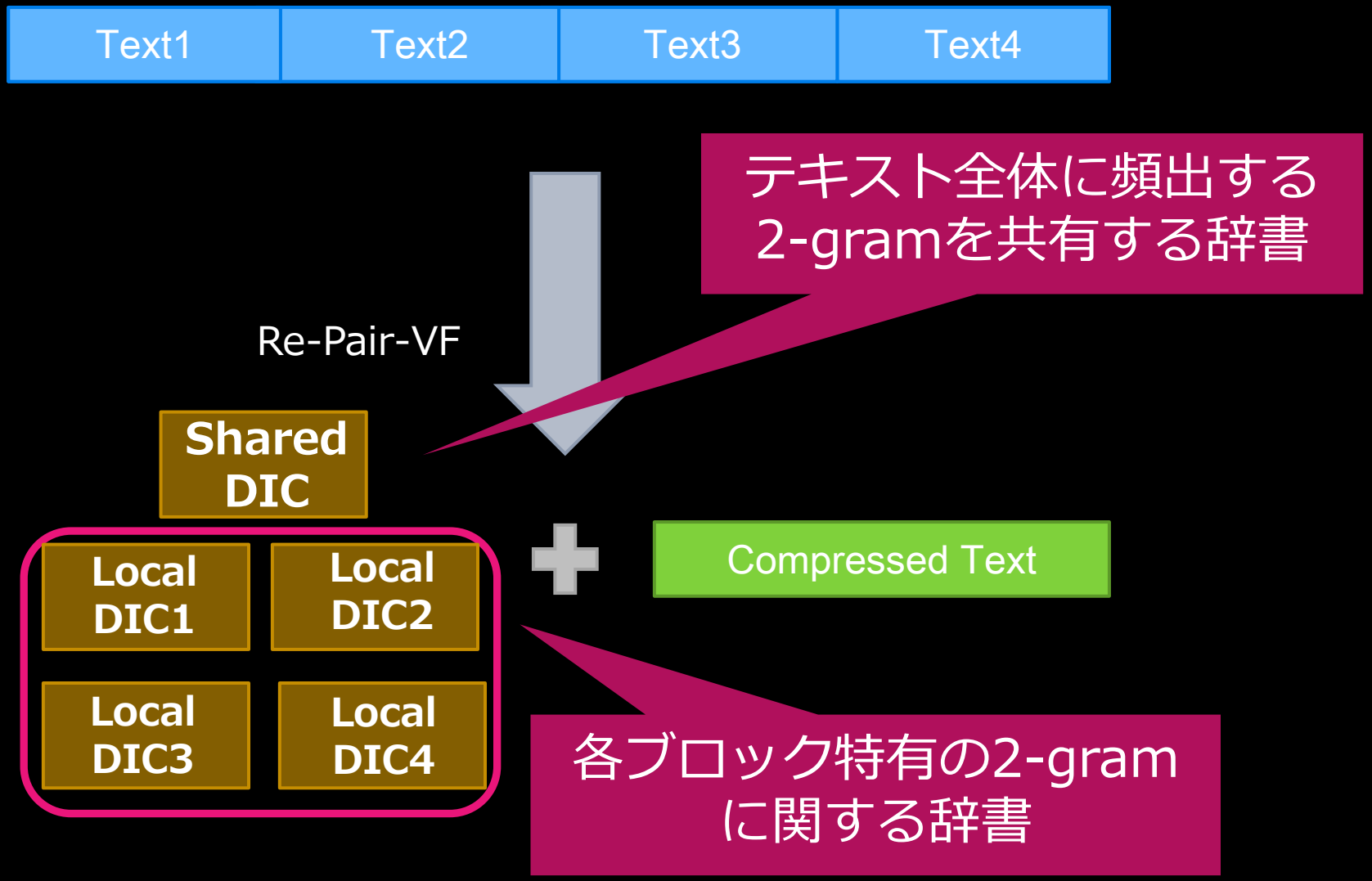
テキストを分割
（ブロック化）



省メモリ化

出力される辞書の一部が同じエントリを持ってしまう．
（予備実験では，各ブロックで30%程度の2-gramの重複を確認）

対策：辞書の共有化


Text1
Text2
Text3
Text4
テキスト全体に頻出する
2-gramを共有する辞書
Re-Pair-VF
Shared DIC
Local DIC1
Local DIC2
Local DIC3
Local DIC4
Compressed Text
各ブロック特有の2-gram
に関する辞書

研究目的と主結果

目的

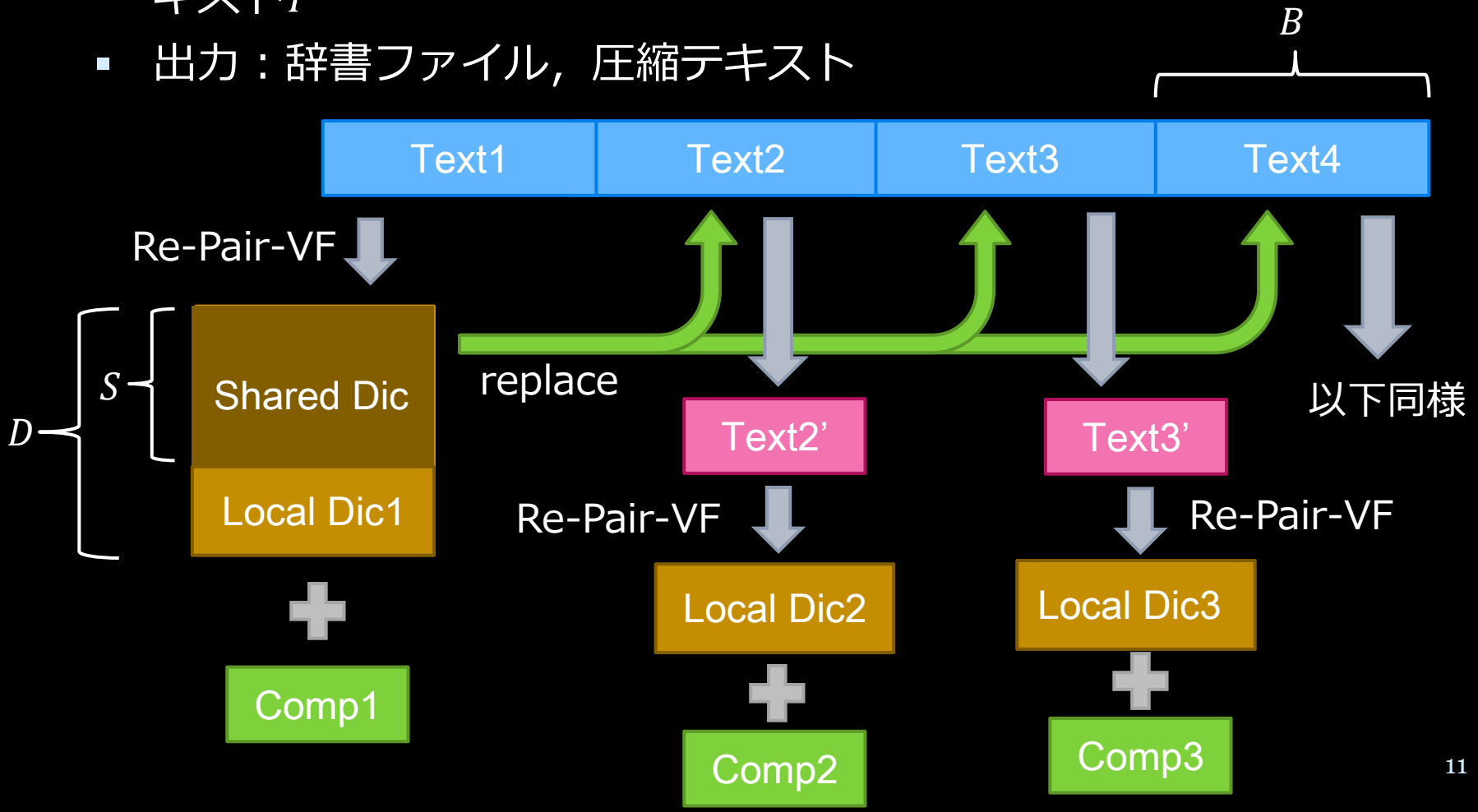
- Re-Pair-VFに対し，入力テキストのブロック化と辞書の共有化を行った際の効果を調査する.

結果

- 改良アルゴリズムBlocked-Re-Pair-VFを考案，計算機実験によりその有用性を示した.
  - 辞書の共有化により圧縮率と圧縮時間がわずかに改善.
  - 省メモリ化により大規模テキストに適用可能に.
  - bzip2に匹敵する圧縮率（約30%）を達成.

提案手法：Blocked-Re-Pair-VF

- 入力：辞書サイズD，共有辞書サイズS，ブロックサイズB，テキストT
- 出力：辞書ファイル，圧縮テキスト

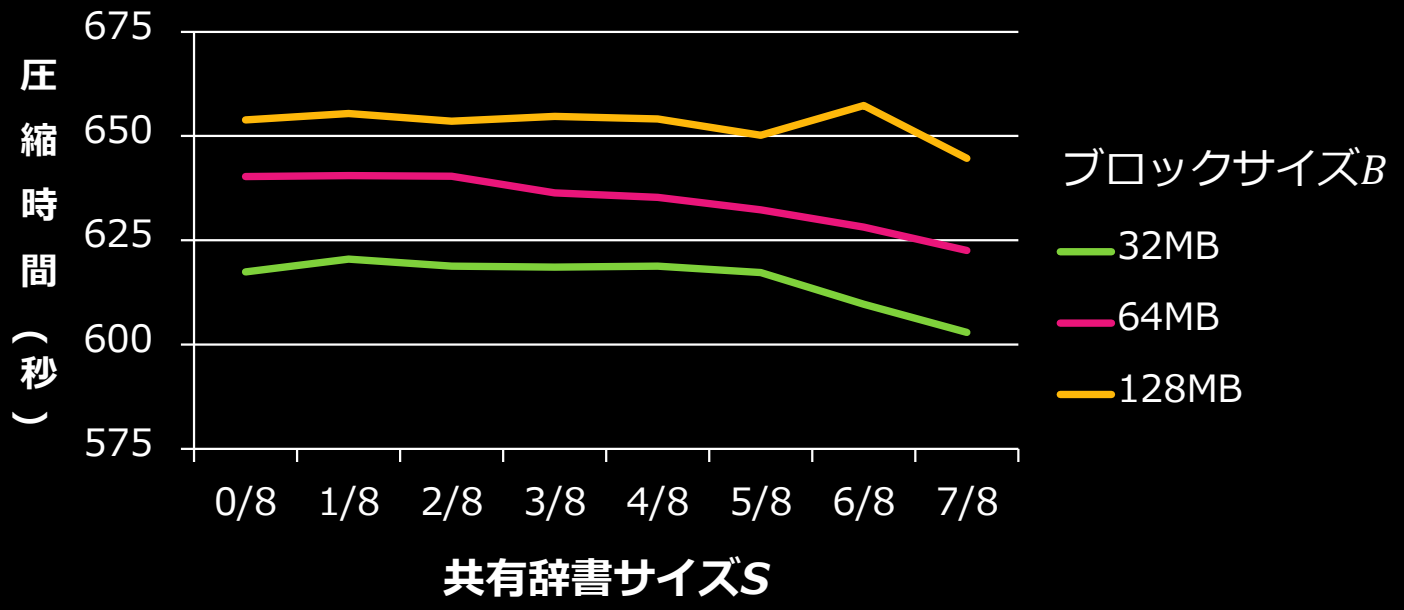


実験 1

- 目的
  - Blocked-Re-Pair-VF において，ブロック化による実行時メモリ消費および辞書の共有化による圧縮パフォーマンス（圧縮率，圧縮時間）を調査する.
- 方法
  - Blocked-Re-Pair-VF において，辞書サイズDと，共有辞書サイズS，ブロックサイズBの3つのパラメータを変化させながら，圧縮時間と圧縮率を計測.
- データ
  - 大きさ約2.2GB，アルファベットサイズ239の英文テキストデータ（ “english” from Pizza & Chili corpus）

実験１-1：辞書の共有化による効果

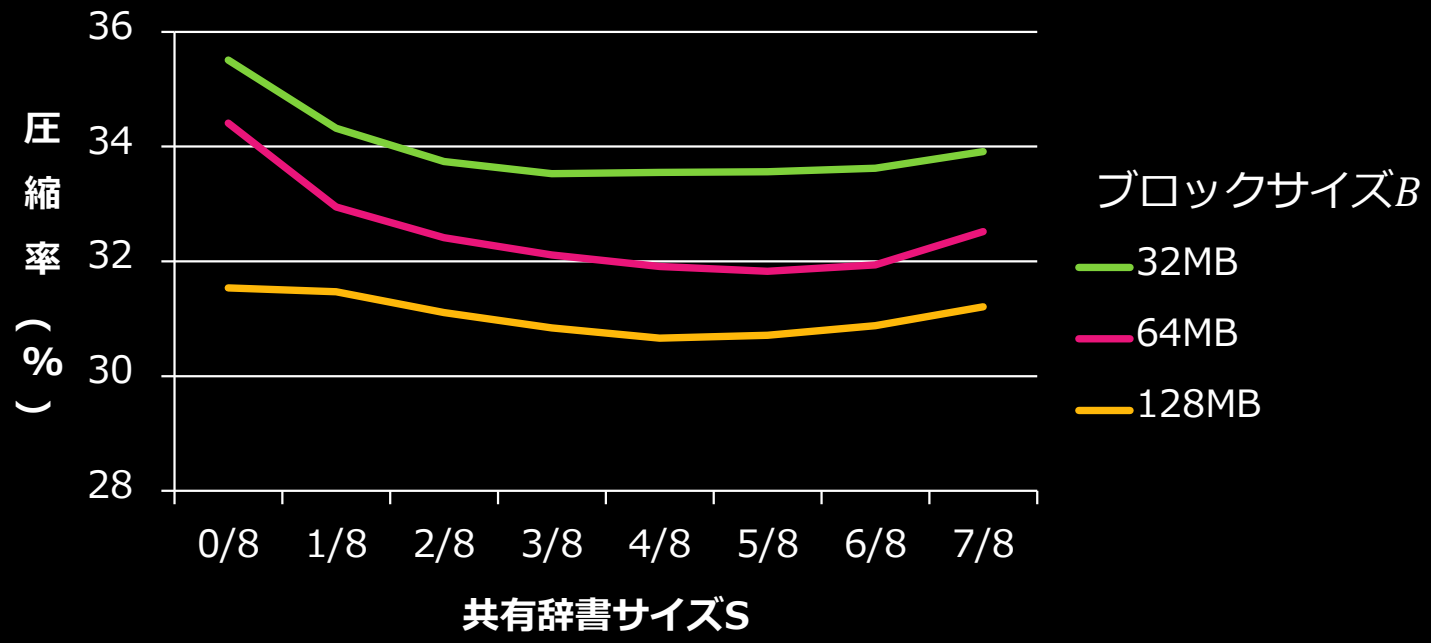
- 圧縮時間の変化
  - 辞書サイズ $D = 2^{20}$



- 辞書の共有化によりわずかに圧縮時間が改善
- ローカル辞書生成時間が短くなるため

実験１-1：辞書の共有化による効果

- 圧縮率の変化
  - 辞書サイズ $D = 2^{20}$



- 辞書の共有化によりわずかな改善が見られた

実験１-2：メモリ消費量の比較

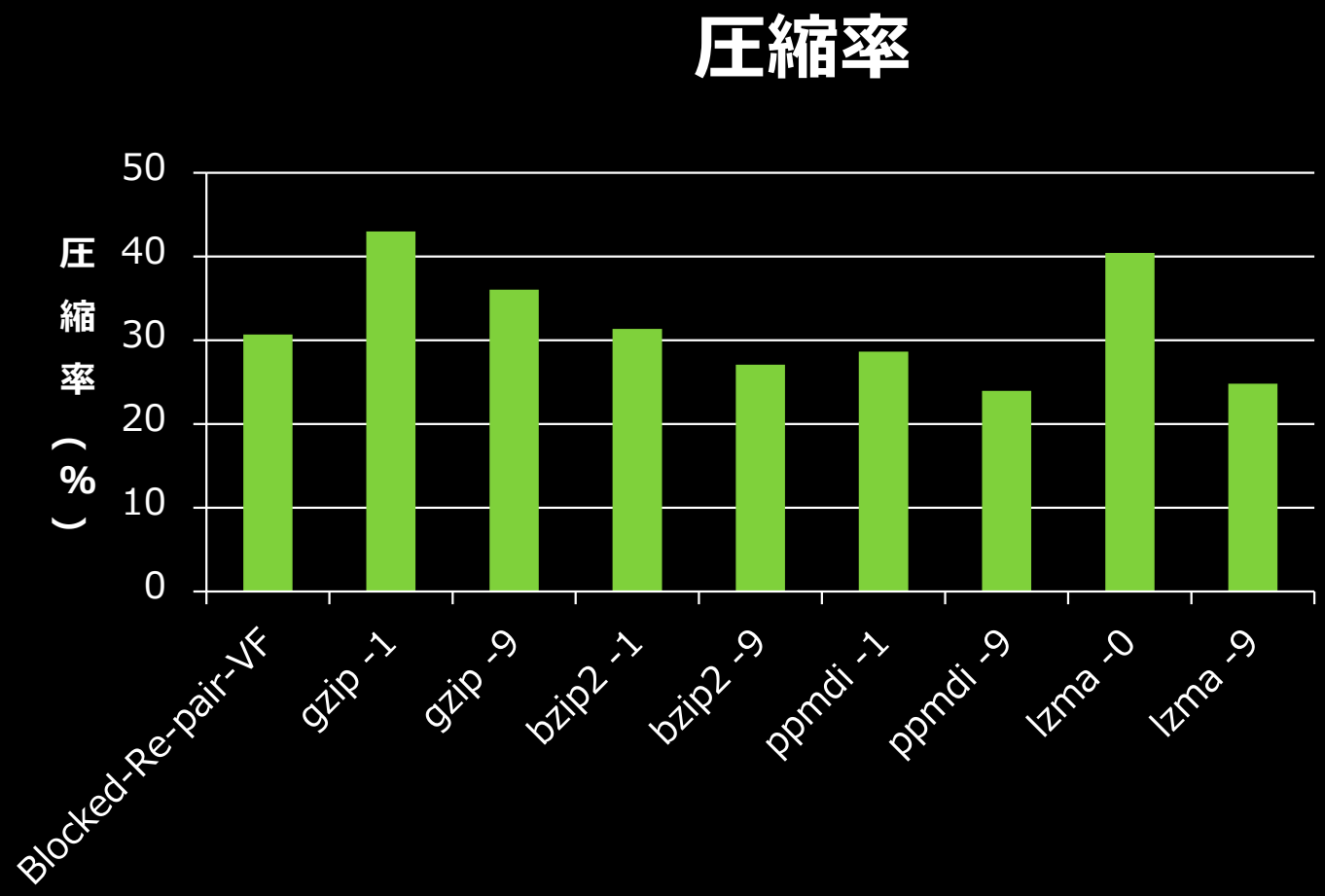
- Re-Pair-VF
  - データ：englishの先頭200MB
  - 圧縮率 (%)：30.9
  - 圧縮時間 (秒)：62.13
  - メモリ消費 (MB)：2656.256
- Blocked-Re-Pair-VF
  - 辞書サイズ $D = 2^{20}$,共有辞書サイズ $S = 1/2$
  - データ：english 全体（約2.2GB）

ブロックサイズB (MB)	32	64	128
圧縮率(%)	33.55	31.91	30.66
圧縮時間(秒)	618.8	635.3	654.1
メモリ消費（MB）	627	1207	2358

実験２

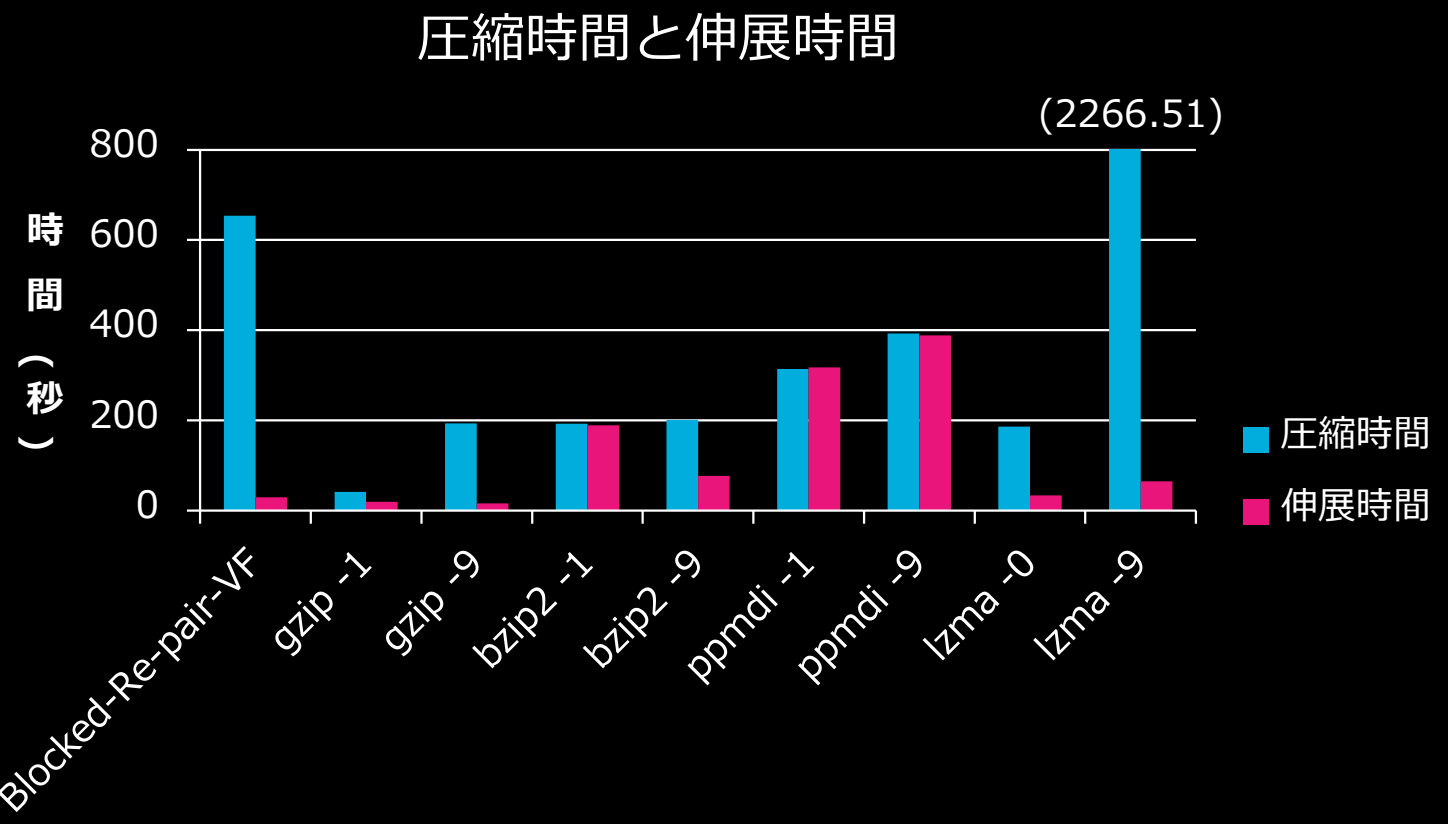
- 目的
  - 既存の圧縮手法とBlocked-Re-Pair-VFの圧縮パフォーマンス（圧縮率，圧縮時間，伸展時間）を比較する.
- 手法
  - Blocked-Re-Pair-VF（提案手法）
  - gzip
  - bzip2
  - ppmd
  - lzma
- データ
  - 大きさ約2.2GB，アルファベットサイズ239の英文テキストデータ（“english” from Pizza & Chili corpus）

実験２：既存手法との比較



提案手法はbzip2に匹敵する圧縮率（約30%）を達成.
（このとき $D = 2^{20}$, $B = 128\text{MB}$, $S = 1/2$）

実験２：既存手法との比較



- 提案手法は，圧縮時間は遅いが，伸展時間は速い.
- 伸展せずに処理を行うことを仮定すれば既存手法と同等の有用性がある.

まとめ

- 結果
  - 大規模テキストに対して，VF符号による圧縮を行うためのアルゴリズムであるBlocked-Re-Pair-VFを提案.
  - 辞書の共有化によって，圧縮率が5%程度改善した.
  - 圧縮率をほとんど悪化させずメモリ消費量を低減.
  - 適切なパラメータにおいて，bzip2並の圧縮率を達成.
- 今後の展望
  - 適切なパラメータの動的な決定法を提案.
  - 共有辞書の作成方法の工夫.