# 決定的アルゴリズムによる単語分散表現の離散符号化

仲村 祐希[1]　鈴木 潤[1,2]　高橋 諒[1,2]　乾 健太郎[1,2]

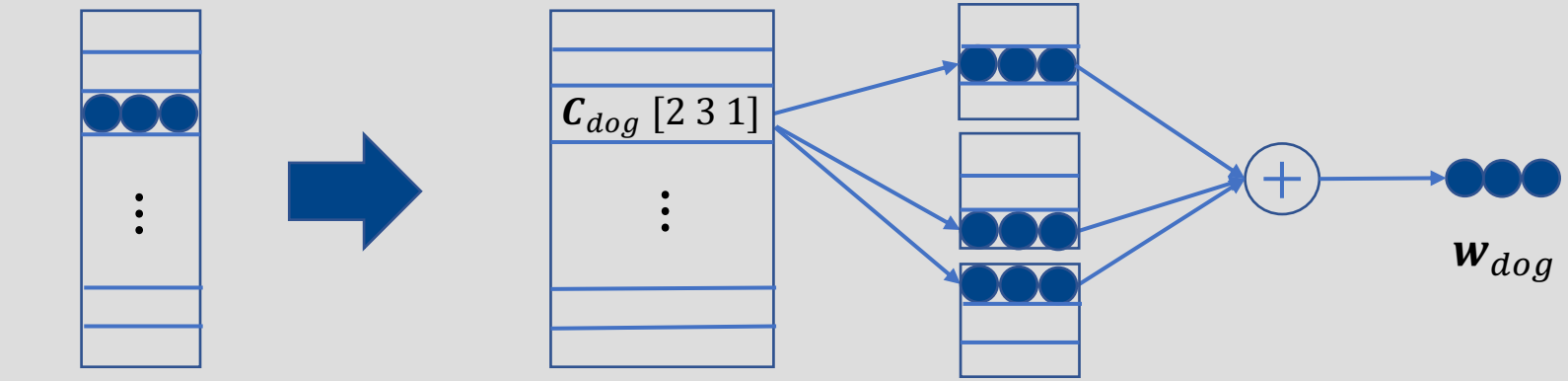
東北大学[1]

理化学研究所[2]

# 単語分散表現の圧縮：基底番号のリスト(離散符号)と基底ベクトルで表現

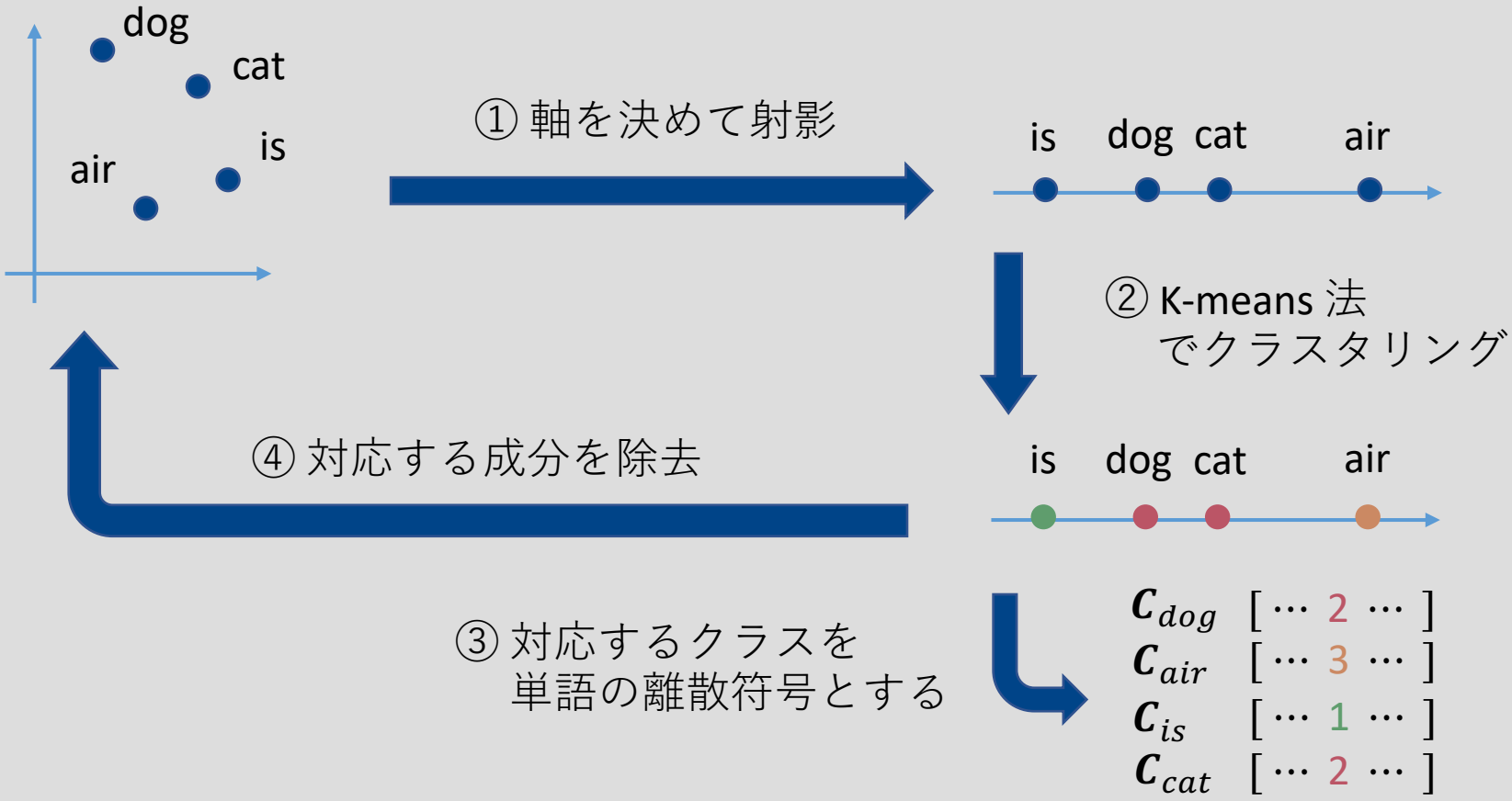
- 深層ニューラルネットワーク(DNN)による手法
  - Compressing Word Embeddings via Deep Compositional Code Learning [Shu+,ICLR'18]
- DNNによる手法は**ランダム性があり**，乱数のシードによって離散符号が**異なる**

➤ **ランダム性がない離散符号の獲得手法を考案**



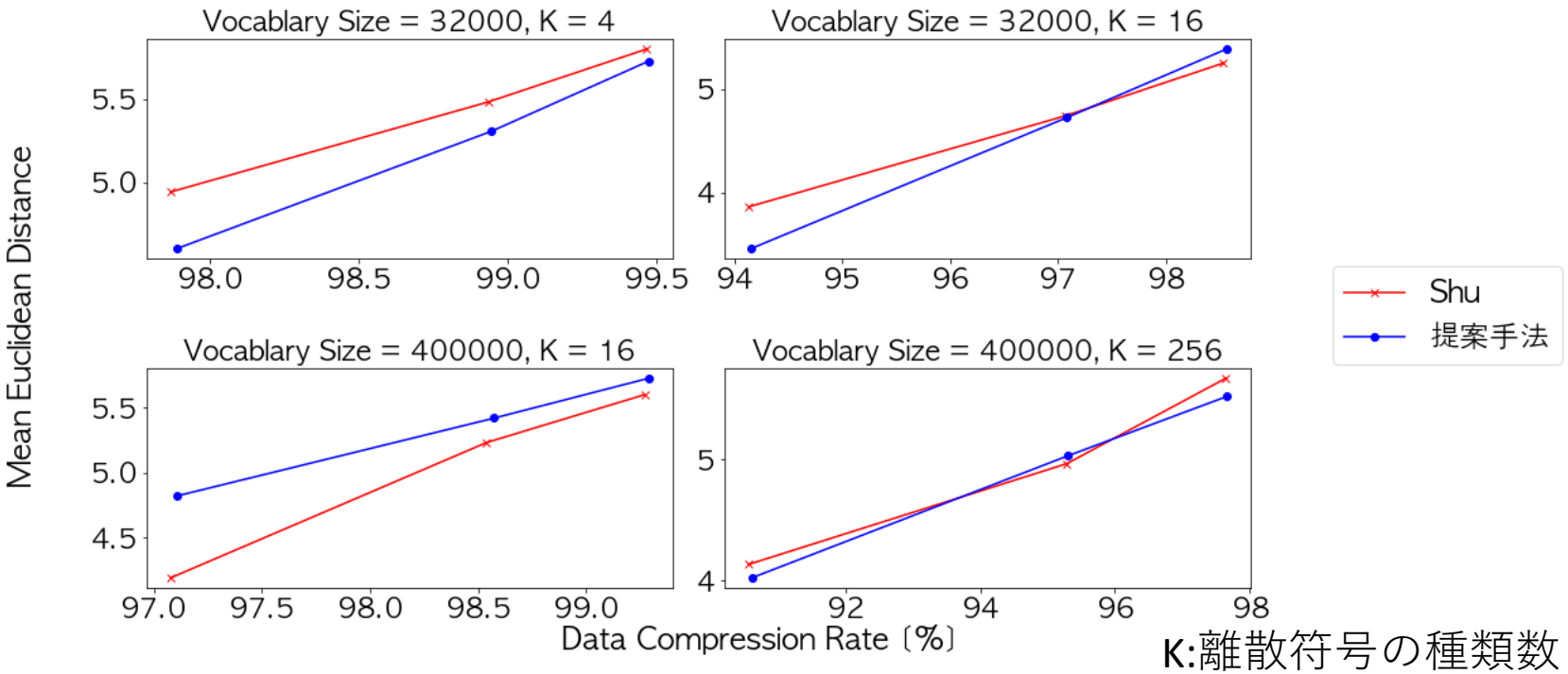
# 提案手法：決定的アルゴリズムによる離散符号の獲得手法

- 1次元のK-means法は最適解が多項式時間で求まり**決定的**



# 結果①：既存手法 [Shu+,ICLR'18]との比較

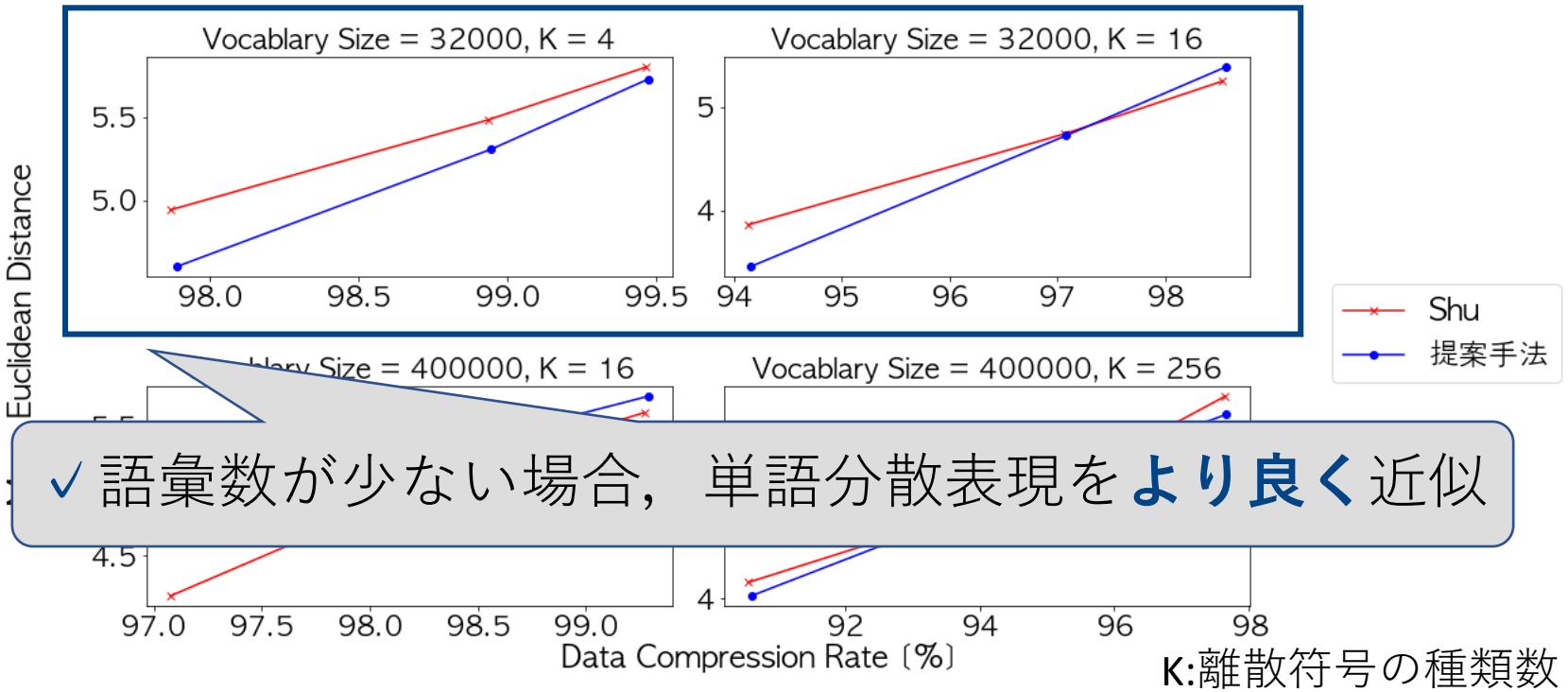
- 圧縮率を揃えて，単語分散表現の圧縮前後の**誤差**を測るためにユークリッド距離を測定




K:離散符号の種類数

# 結果①：既存手法 [Shu+,ICLR'18]との比較

- 圧縮率を揃えて，単語分散表現の圧縮前後の**誤差**を測るためにユークリッド距離を測定



# 結果②：離散符号例

- 既存手法 [Shu+,ICLR'18]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1回目 | dog | 5 | 10 | 7 | 11 | 4 | 1 | 12 | 14 |
| | dogs | 6 | 5 | 7 | 1 | 4 | 1 | 12 | 3 |
| 2回目 | dog | 7 | 5 | 15 | 2 | 7 | 15 | 3 | 3 |
| | dogs | 7 | 5 | 6 | 7 | 7 | 8 | 11 | 3 |
| 3回目 | dog | 9 | 4 | 11 | 11 | 0 | 11 | 1 | 2 |
| | dogs | 9 | 4 | 3 | 0 | 0 | 11 | 1 | 2 |

- 提案手法

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1回目 | dog | 11 | 3 | 3 | 12 | 4 | 4 | 7 | 8 |
| | dogs | 8 | 3 | 6 | 14 | 3 | 4 | 10 | 9 |
| 2回目 | dog | 11 | 3 | 3 | 12 | 4 | 4 | 7 | 8 |
| | dogs | 8 | 3 | 6 | 14 | 3 | 4 | 10 | 9 |
| 3回目 | dog | 11 | 3 | 3 | 12 | 4 | 4 | 7 | 8 |
| | dogs | 8 | 3 | 6 | 14 | 3 | 4 | 10 | 9 |

✓提案手法は乱数のシードを変えても離散符号は**不変**

# まとめと議論

- **決定的アルゴリズム**による単語分散表現の離散符号化手法を考案した

**議論**

- さらなる性能向上のための手法やDNNの中に取り入れるための工夫
- 決定的な離散符号の他の適用先の検討

## Contact

- https://yukinon874.github.io/

# Appendix

# Appendix：まとめと議論

- **決定的アルゴリズム**による単語分散表現の離散符号化手法を考案した

- 機械翻訳タスクなどに適用した場合，**性能を落とさず**にどの程度まで**圧縮**できるか測定したい

**議論**

- さらなる性能向上のための手法やDNNの中に取り入れるための工夫
- 決定的な離散符号の他の適用先の検討

# Appendix：軸の決め方の詳細

- 主成分分析 (実験ではこちらを使用)
  - **分散が最大**となるように軸を決定

- 単語の類似度行列を二つの1次元のベクトルの積で近似
  - Right-truncatable Neural Word Embeddings [Suzuki+,NAACL'16]

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{W}^T$$

$$\text{Minimize} \ \frac{1}{2}\sum_{(i,j)}\left(x_{i,j} - u_i v_j\right)^2$$

$\boldsymbol{W}$:単語分散表現
$\boldsymbol{u}^{V\times 1}$:1次元近似ベクトル
$\boldsymbol{v}^{1\times V}$:1次元近似ベクトル
$V$:語彙数

  - 1次元近似ベクトルの片方をクラスタリングに用いる

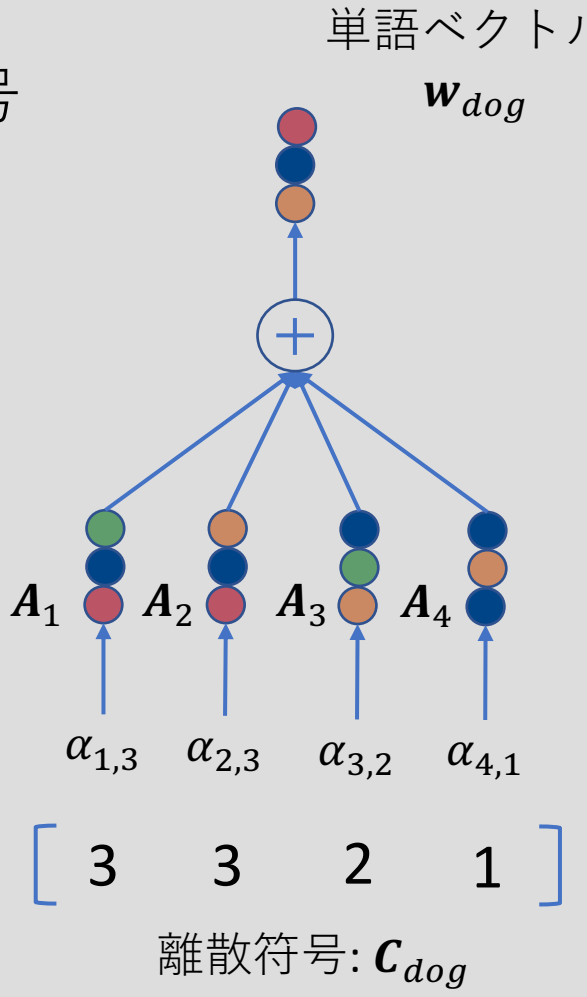
# Appendix：実験で用いた基底ベクトルの詳細

- 単語分散表現を基底ベクトルと離散符号に対応する基底ベクトルの重みで表現

$$\boldsymbol{w} = \sum_i \alpha_{i,C_i} \boldsymbol{A}_i$$

$\boldsymbol{w}$:単語ベクトル
$\boldsymbol{C}$:離散符号
$\alpha$:基底ベクトルの重み
$\boldsymbol{A}$:基底ベクトル

- 単語分散表現との平均二乗誤差が最小となるようにDNNで学習



# Appendix：手法によるサイズの違い

- 既存手法 [Shu+,ICLR'18]

$$VM \log K + 4MKH \text{ [Byte]}$$

M:離散符号の数
K:離散符号の種類数
H:単語ベクトルの次元数
V:語彙数

- 提案手法

$$VM \log K + 4MH + 4MK \text{ [Byte]}$$

- 圧縮率

$$\frac{\text{圧縮前のサイズ} - \text{圧縮後のサイズ}}{\text{圧縮前のサイズ}} [\%]$$