

事前学習済み Transformer を用いた Data-to-text における 入力順序の影響分析

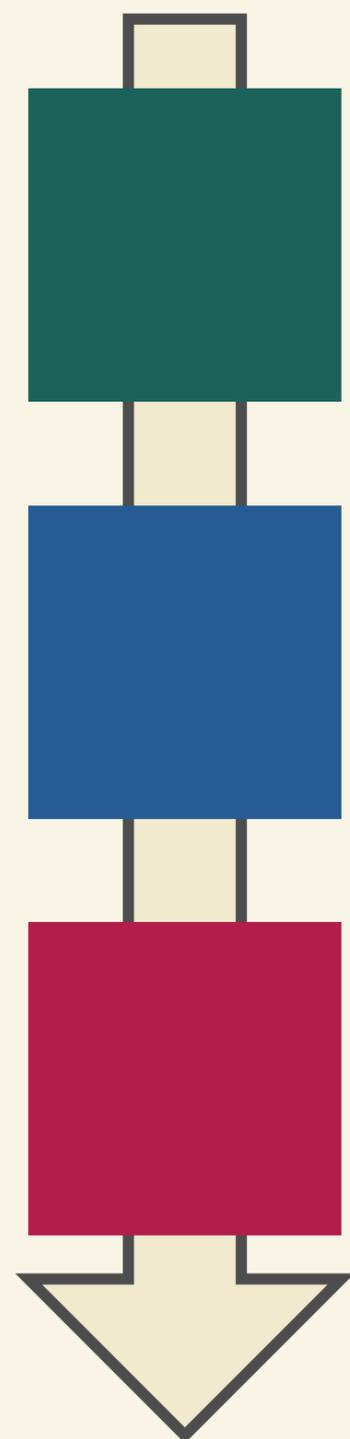
奈良先端科学技術大学院大学

矢野祐貴

須藤克仁

中村 哲

発表の流れ



Data-to-textと関連研究 について

研究・実験の目的，実験設定 について

実験結果と解析，その考察 について

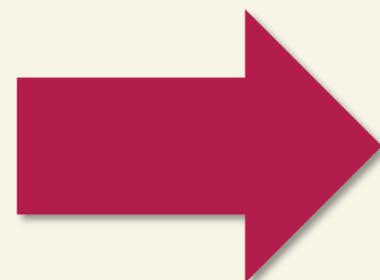
Data-to-text について

Data-to-text とは

構造化データ から 要約文/説明文 を生成するタスク

- 構造化データ：表やグラフ など

Slot	Value
場所	大阪
気温	36
日時	あす
天気	晴れ



あすの大阪は晴れで、
気温は36°Cまで上がる見込みです

出力：説明文

入力：構造化データ(表)

構造化データ について

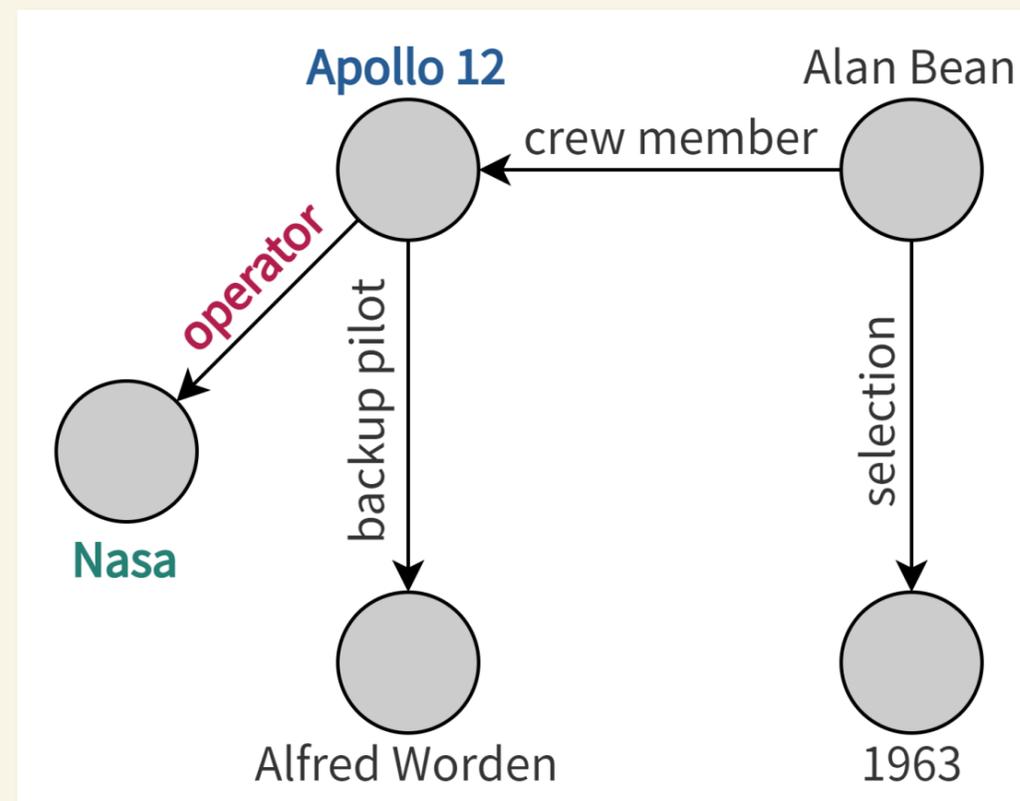
構造化データは 「レコードの集合」と定式化

$$\mathcal{X} = \text{set}\{r_1, r_2, \dots, r_i, \dots, r_n\}$$

表 $r_i = (\text{slot}_i, \text{value}_i)$

Slot	Value
場所	大阪
気温	36
日時	あす
天気	晴れ

グラフ $r_i = (\text{Head}_i, \text{Relation}_i, \text{Tail}_i)$



構造化データの特徴量化手法

1. 入力を線形化して埋め込み

メリット

データの形式に依存しない手法

語彙化のような前/後処理や、
事前知識を活用しなければ、
まともな生成が困難

2. 構造を考慮してエンコード

設計によっては、
より豊富な特徴を獲得可能
自然な手法

データごとの設計が必要

デメリット

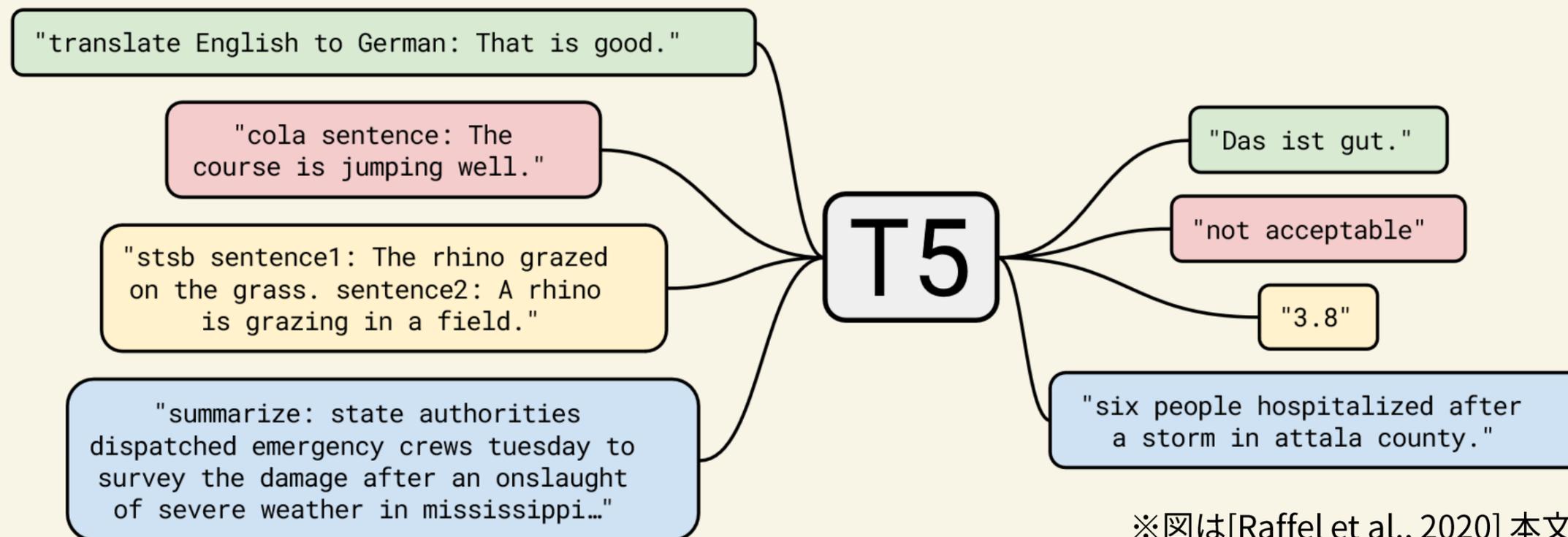
関連研究について

関連研究① [Kale et al., 2020] T5 を用いた Data-to-text の研究

† “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, Raffel et al., 2020

T5: Text-to-text Transfer Transformer †

- 要約, 翻訳, 質問応答生成 (Text-to-text) タスクを同一モデルで解く



※図は[Raffel et al., 2020]本文中より引用

Data-to-textでは...

- 入力の線形化によってText-to-textとみなして同様に生成

T5 を用いた Data-to-text の研究

当時のState-of-the-artを達成

- 3つのデータセット：WebNLG, MultiWoz, ToTTo

これらのデータの特徴

- 出力が **入力のほぼ全てのレコード** について言及している (例↓)

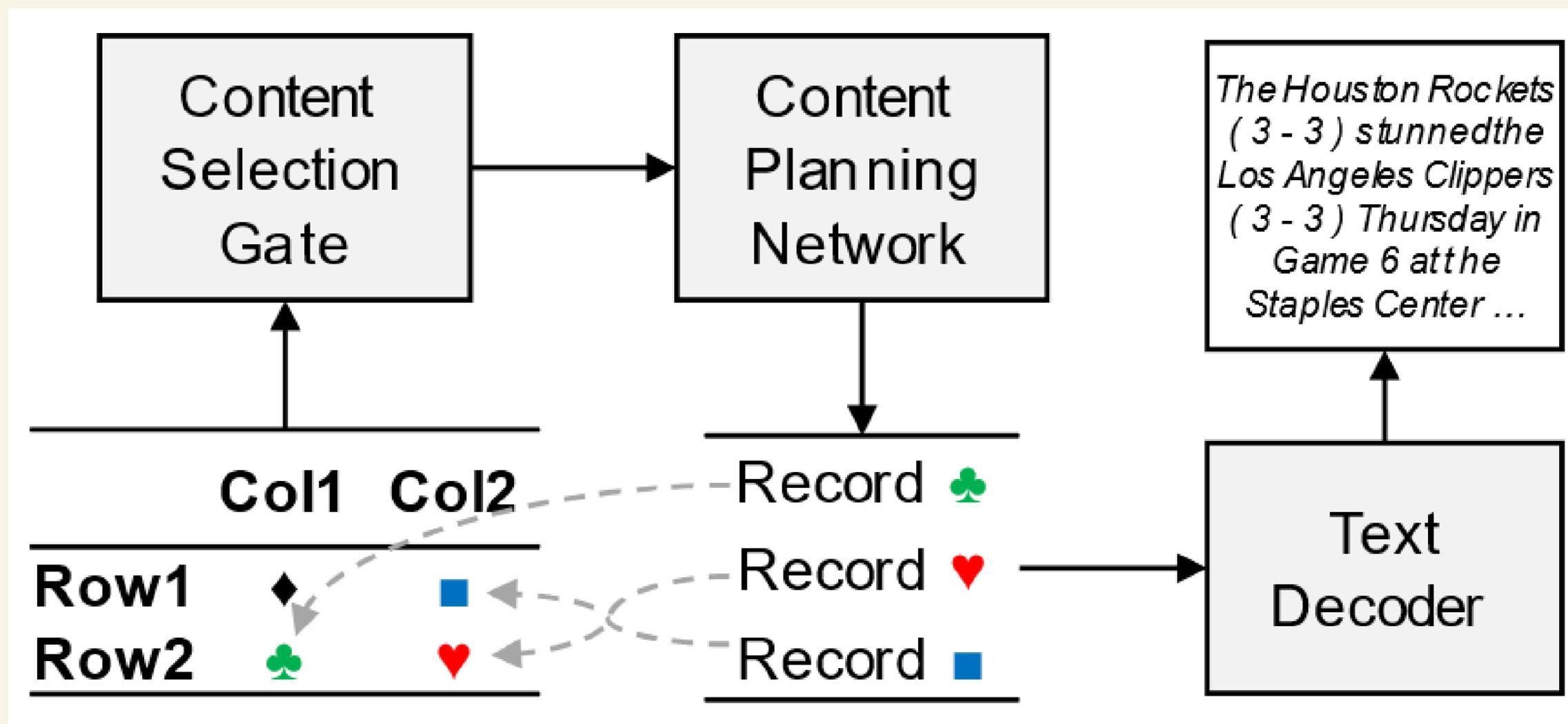
Slot	Value
場所	大阪
気温	36
日時	あす
天気	晴れ



あすの大阪は晴れで、
気温は36°Cまで上がる見込みです

選択とプランニングによる Data-to-text

「何を言うか」, 「どの順序で言うか」を学習



※図は[Puduppully et al., 2020]本文中より引用

選択とプランニングによる Data-to-text

モデルは注意付きRNNモデルベース

- 特徴量化手法は「構造を考慮したエンコード」

RotoWireデータセットで精度向上

- 入力の一部のレコードにのみ言及（例↓）

	あす	あさって	...
大阪	はれ / 36°C	くもり / 31°C	...
神戸	くもり / 30°C	あめ / 26°C	...
⋮	⋮	⋮	



あすの大阪は晴れで、
気温は36°Cまで上がる見込みです。
神戸ではあさって、あめが降るでしょう。

関連研究のまとめ

関連研究① T5を用いた Data-to-text

関連研究② 選択と**プランニング**による Data-to-text

アーキテクチャ

事前学習済みTransformer

注意付きRNNモデル

入力の特徴量

入力を線形化して埋め込み

構造に基づいてエンコード

データセット
の特徴

全てのレコードに言及

一部のレコードにのみ言及

本研究のベースライン

T5 を用いた Data-to-text における 入力順序の影響分析 について

本研究の目的

関連研究①

本研究のベースライン

事前学習済みTransformer

かつ

全てのレコードに言及

有効かどうかを検証する

関連研究②

レコード順序のプランニング

関連研究① (T5によるData-to-text) では、 レコード順序をどう扱っているか？

データセットの そのままの順序で線形化

- レコード順序は考慮していない
- 何らかの人為的整列が予想

※WebNLG学習データから抜粋

```

1 <modifiedtripleaset>↓
2 <mtriple>Visvesvaraya_Technological_University | city | Belgaum</mtriple>↓
3 <mtriple>Acharya_Institute_of_Technology | director | "Dr. G. P. Prabhukuma
4 <mtriple>Acharya_Institute_of_Technology | established | 2000</mtriple>↓
5 <mtriple>Acharya_Institute_of_Technology | country | "India"</mtriple>↓
6 <mtriple>Acharya_Institute_of_Technology | numberOfPostgraduateStudents | 7
7 <mtriple>Acharya_Institute_of_Technology | campus | "In Soldevanahalli, Ach
8 <mtriple>Acharya_Institute_of_Technology | affiliation | Visvesvaraya_Techn
9 </modifiedtripleaset>↓
10
11 <modifiedtripleaset>↓
12 <mtriple>Visvesvaraya_Technological_University | city | Belgaum</mtriple>↓
13 <mtriple>Acharya_Institute_of_Technology | established | 2000</mtriple>↓
14 <mtriple>Acharya_Institute_of_Technology | motto | "Nurturing Excellence"<
15 <mtriple>Acharya_Institute_of_Technology | state | Karnataka</mtriple>↓
16 <mtriple>Acharya_Institute_of_Technology | country | "India"</mtriple>↓
17 <mtriple>Acharya_Institute_of_Technology | campus | "In Soldevanahalli, Ach
18 <mtriple>Acharya_Institute_of_Technology | affiliation | Visvesvaraya_Techn
19 </modifiedtripleaset>↓

```

Q1. T5モデルは、「何らかの人為的整列」を学習しているか？

Q2. T5モデルは、集合に近いレコード入力
(シャッフルした入力) をどう学習するか？ **15**

関連研究① (T5によるData-to-text) では、 レコード順序をどう扱っているか？

データセットの そのままの順序で線形化

- レコード順序は考慮していない
- 何らかの人為的整列が予想

※WebNLG学習データから抜粋

```

1 <modifiedtriple>↓
2 <mtriple>Visvesvaraya Technological University | city | Belgaum</mtriple>↓
3 <mtriple>Aacharya_Institute_of_Technology | director | Dr. G. P. Prabhukuma</mtriple>↓
4 <mtriple>Aacharya_Institute_of_Technology | established | 2000</mtriple>↓
5 <mtriple>Aacharya_Institute_of_Technology | country | "India"</mtriple>↓
6 <mtriple>Aacharya_Institute_of_Technology | numberOfPostgraduateStudents | 7</mtriple>↓
7 <mtriple>Aacharya_Institute_of_Technology | campus | "In Soldevanahalli, Ach</mtriple>↓
8 <mtriple>Aacharya_Institute_of_Technology | affiliation | Visvesvaraya_Techn</mtriple>↓
9 </modifiedtriple>↓
10
11 <modifiedtriple>↓
12 <mtriple>Visvesvaraya Technological University | city | Belgaum</mtriple>↓
13 <mtriple>Aacharya_Institute_of_Technology | established | 2000</mtriple>↓
14 <mtriple>Aacharya_Institute_of_Technology | motto | "Nurturing Excellence"</mtriple>↓
15 <mtriple>Aacharya_Institute_of_Technology | state | Karnataka</mtriple>↓
16 <mtriple>Aacharya_Institute_of_Technology | country | "India"</mtriple>↓
17 <mtriple>Aacharya_Institute_of_Technology | campus | "In Soldevanahalli, Ach</mtriple>↓
18 <mtriple>Aacharya_Institute_of_Technology | affiliation | Visvesvaraya_Techn</mtriple>↓
19 </modifiedtriple>↓

```

Q1. T5モデルは、「何らかの人為的整列」を学習しているか？

Q2. T5モデルは、集合に近いレコード入力
(シャッフルした入力) をどう学習するか？ **16**

実験： 3パターンの生成結果を比較

Def-Def

- 元の順序 で Fine-tuning ➡ 元の順序 で 生成 (テスト)

Def-Shuf

- 元の順序 で Fine-tuning ➡ シャッフルした順序 で 生成

Shuf-Shuf

- シャッフルした順序 で Fine-tuning ➡ シャッフルした順序 で 生成

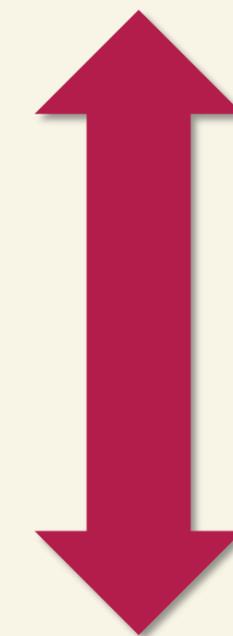
実験： 3パターンの生成結果を比較

Def-Def

- 元の順序 で Fine-tuning ➡ 元の順序 で 生成 (テスト)

Def-Shuf

- 元の順序 で Fine-tuning ➡ シャッフルした順序 で 生成



Q1. T5モデルは、「何らかの人為的整列」を学習しているか？
を検証

実験： 3パターンの生成結果を比較

Def-Def

- 元の順序 で Fine-tuning ➔ 元の順序 で 生成 (テスト)

Q2. T5モデルは、集合に近いレコード入力
(シャッフルした入力) をどう学習するか？
を検証

Shuf-Shuf

- シャッフルした順序 で Fine-tuning ➔ シャッフルした順序 で 生成

実験設定

T5モデルのパラメータサイズ

- 0.6 Mil (Small) / 2.2 Mil (Base) / 7.7 Mil (Large)

Shuf-Shuf について

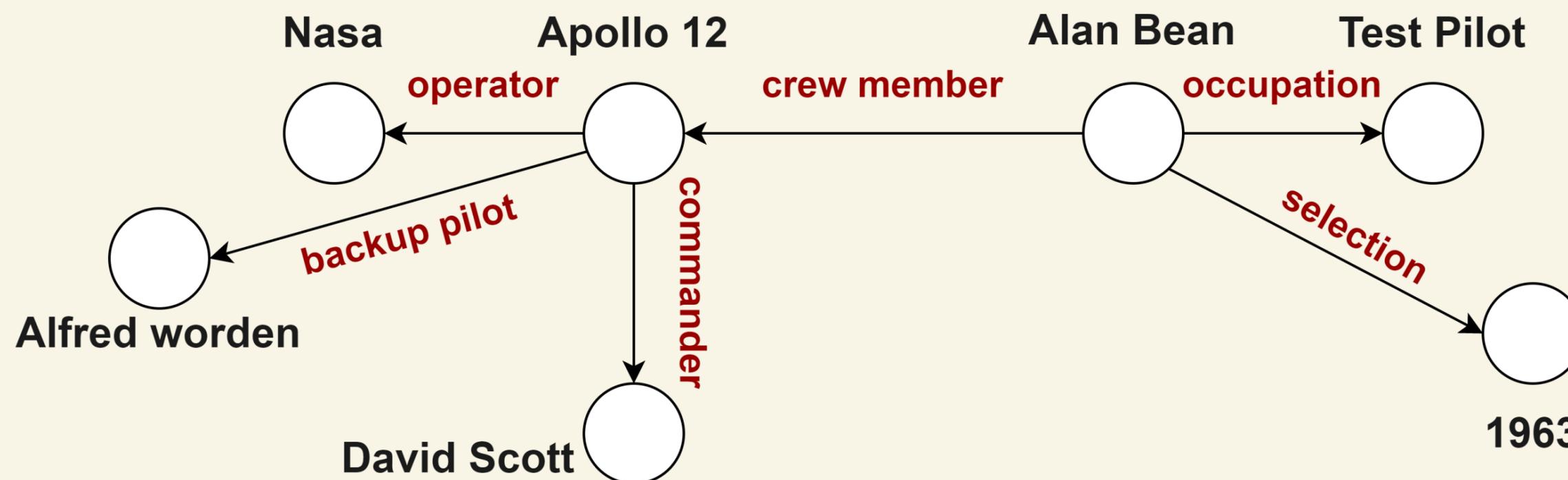
- 実験ごとに **1回のみ** シャッフルを行う
- (エポックごとのシャッフルは今回行わない)

評価指標：BLEU

- エポックの選択
- 生成文の評価 に使用

Webから収集されたグラフとその説明文の組

- 説明文はグラフのすべてのレコードに言及



Alan Bean joined NASA in 1963 where he became a member of the Apollo 12 mission along with Alfred Worden as back up pilot and David Scott as commander.

データのサイズ

- 学習データ: 18.1K 件
- 検証データ: 2.3K 件
- テストデータ: 4.9K 件

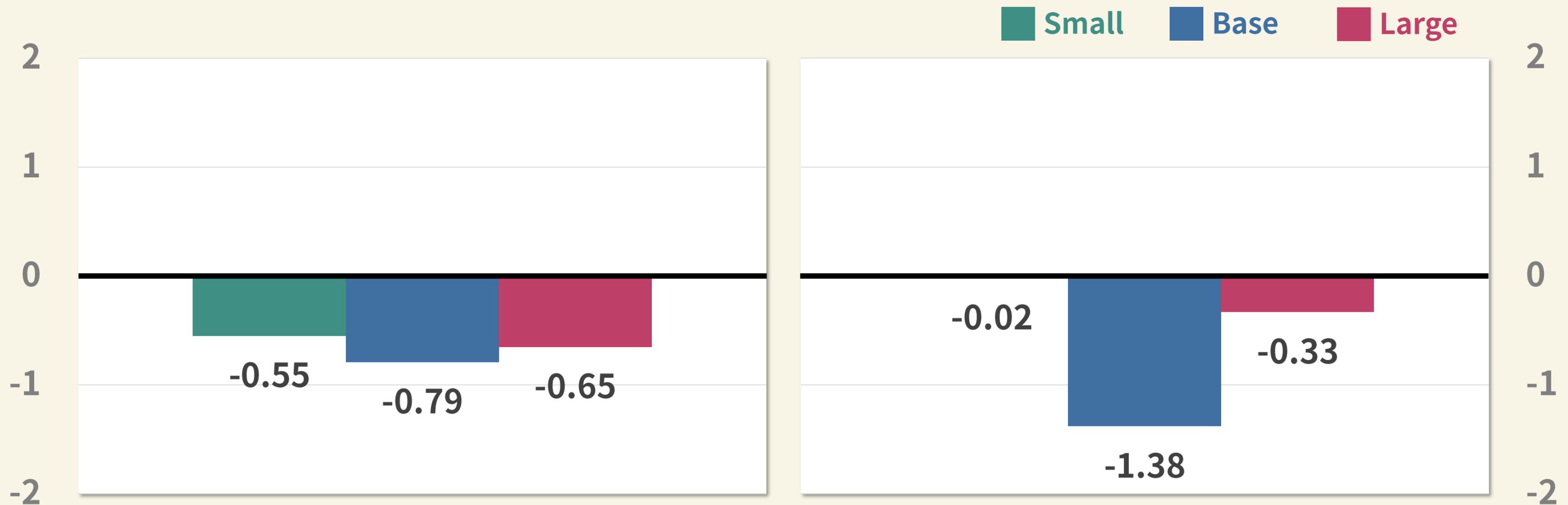
テストデータは更に2分割

- 既知ドメイン: 2.5K 件 (学習データに含まれるドメイン)
- 未知ドメイン: 2.4K 件 (学習データに含まれないドメイン)

実験結果

BLEU値の比較

Def-Shuf (元の順序でFine-tuning **シャッフルした順序**で生成) による変化

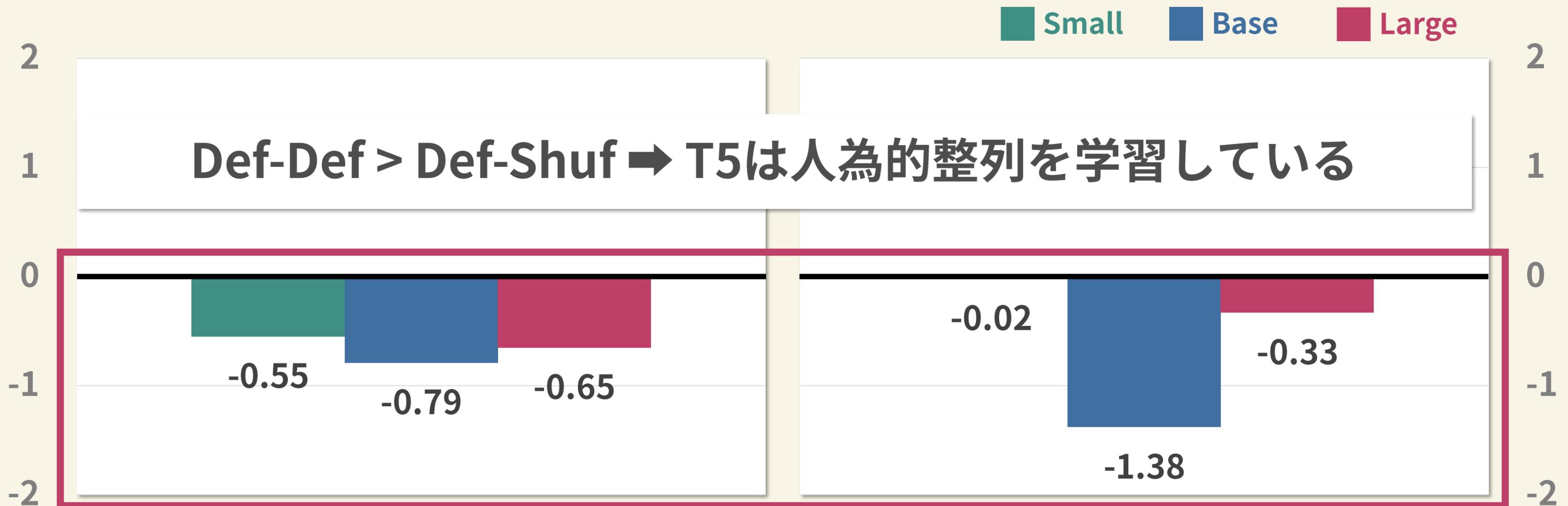


既知ドメイン

未知ドメイン

BLEU値の比較

Def-Shuf (元の順序でFine-tuning **シャッフルした順序**で生成) による変化



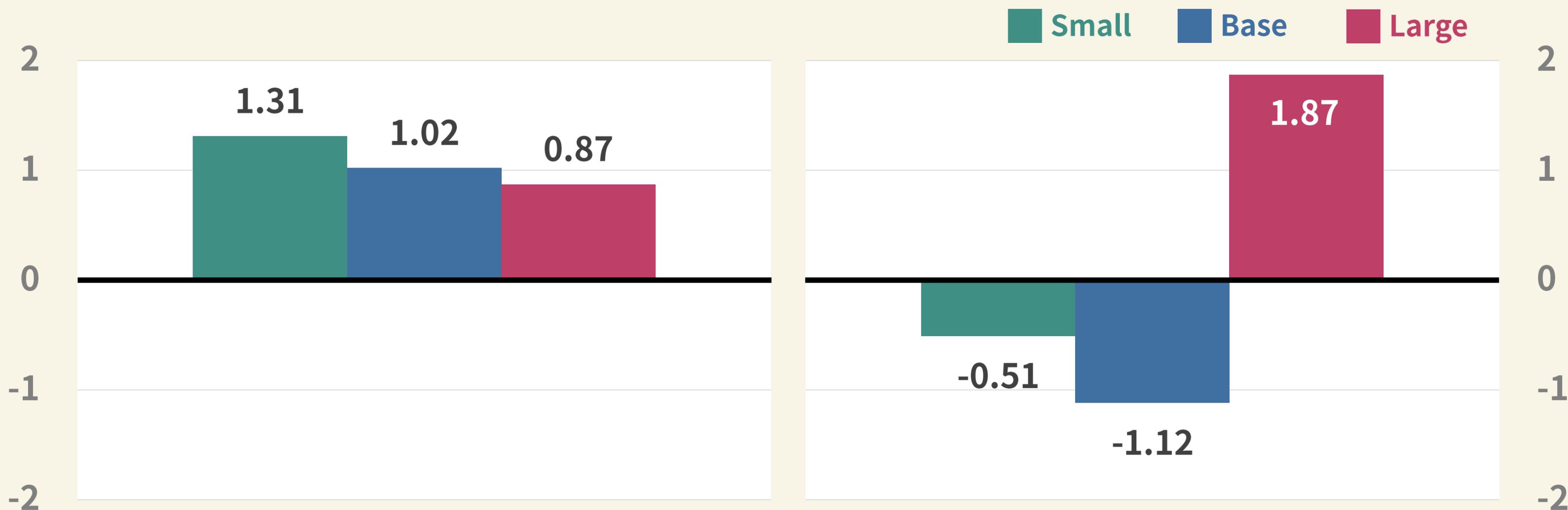
Def-Def > Def-Shuf ⇒ T5は人為的整列を学習している

既知ドメイン

未知ドメイン

BLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化

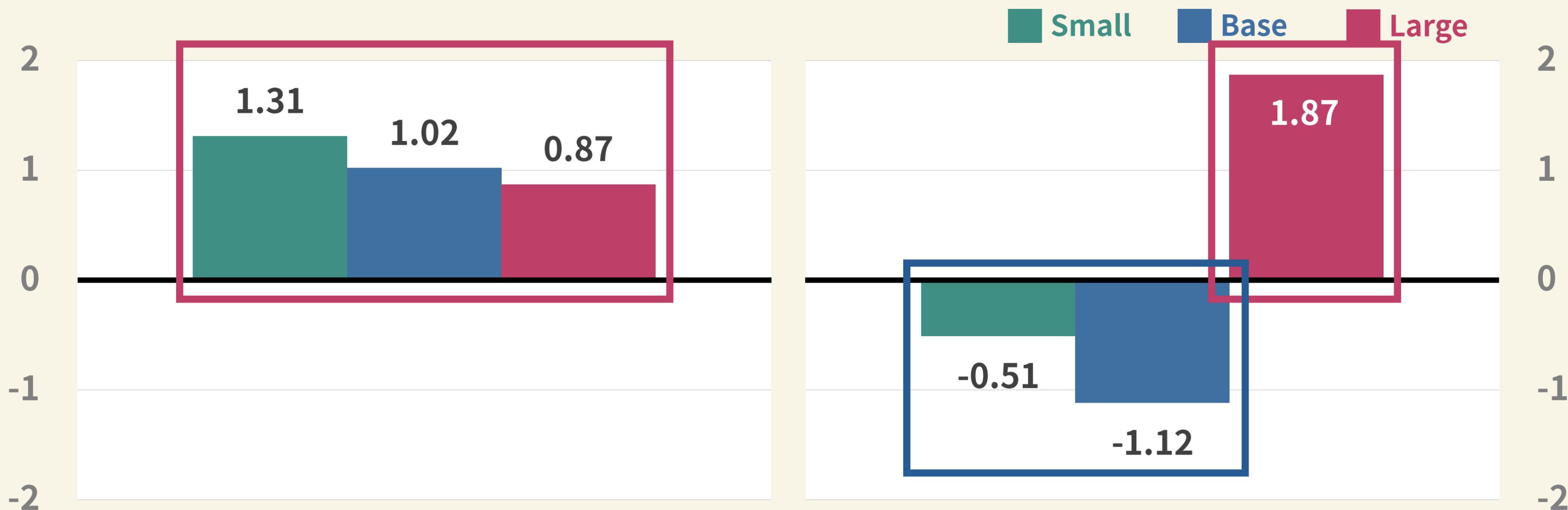


既知ドメイン

未知ドメイン

BLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化

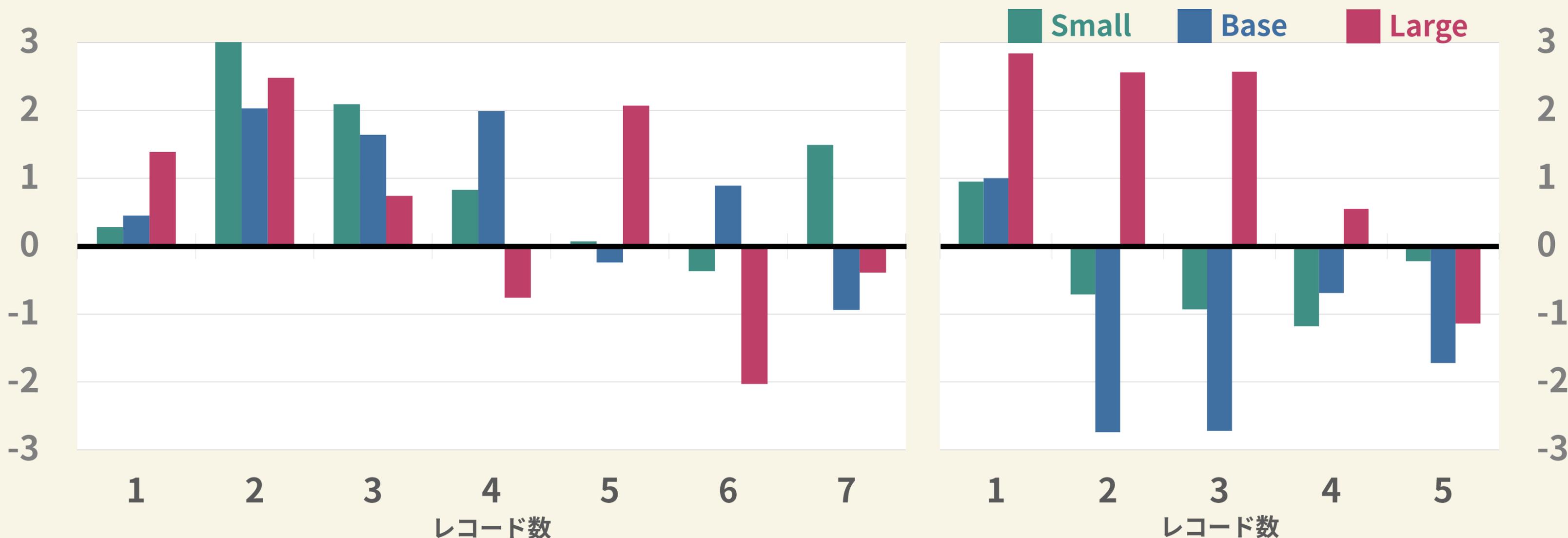


既知ドメイン

未知ドメイン

レコード数ごとのBLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化



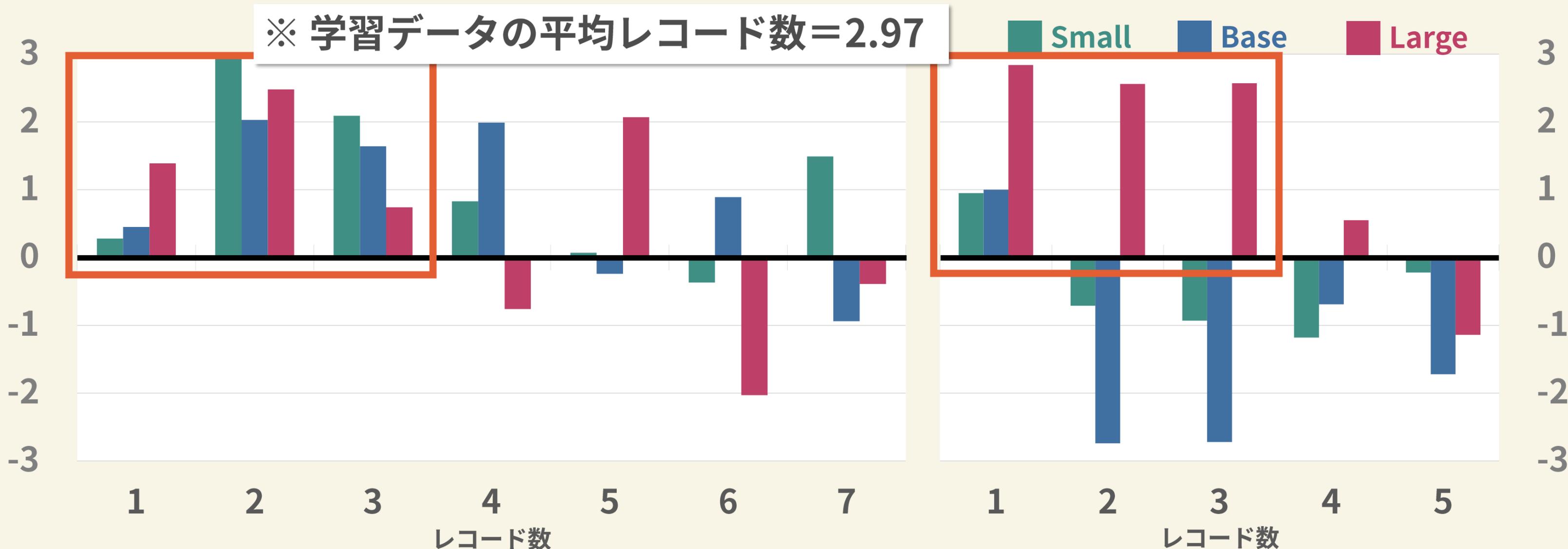
既知ドメイン

未知ドメイン

レコード数ごとのBLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化

※ 学習データの平均レコード数=2.97



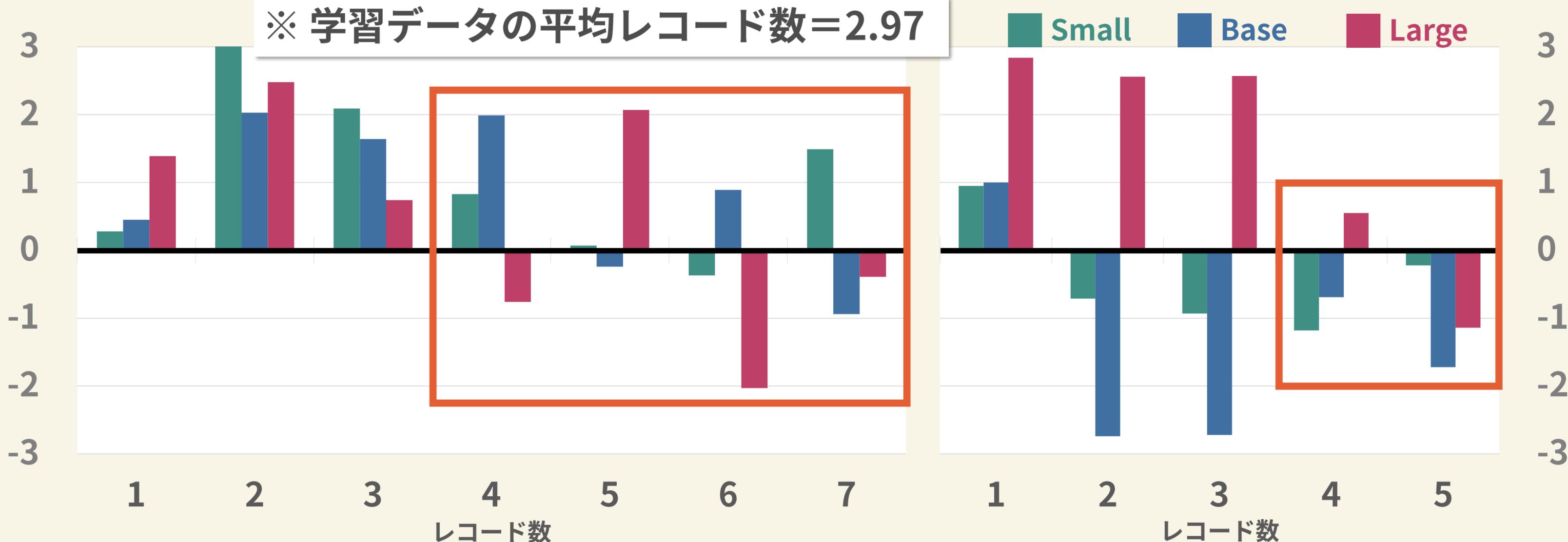
既知ドメイン

未知ドメイン

レコード数ごとのBLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化

※ 学習データの平均レコード数=2.97



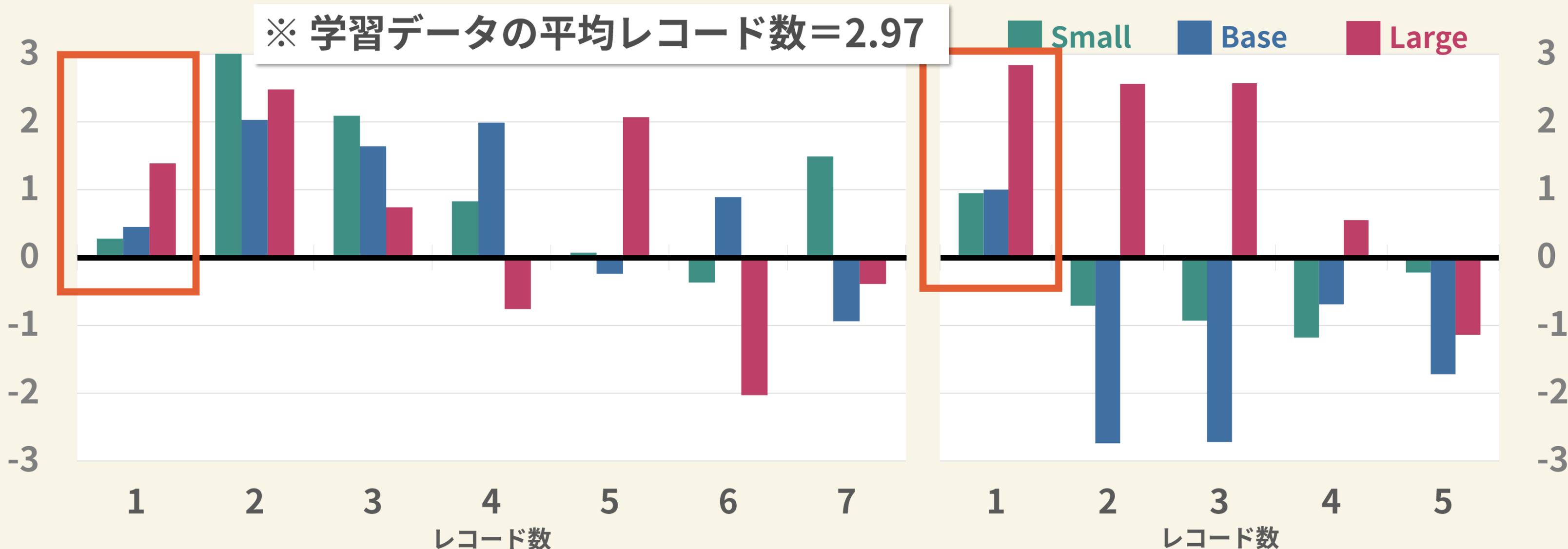
既知ドメイン

未知ドメイン

レコード数ごとのBLEU値の比較

Shuf-Shuf (シャッフルした順序でFine-tuning / 生成) による変化

※ 学習データの平均レコード数=2.97



既知ドメイン

未知ドメイン

検証結果

Q1. 人為的整列を学習しているか？

- 学習している
- 関連研究①の設定でもレコードの順序プランニングは有効そう

Q2. 集合に近い入力をどう学習するか？

- 汎化性能向上に寄与（ただし十分大きなモデルサイズが必要）
- 学習データのレコード数の分布にも依存

➡ シャッフルした順序によるFine-tuningは、**データ拡張**になっている？

目的

- 「T5 かつ全レコードへ言及するデータ」に対しても、「レコード順序のプランニング」が有効かを検証

実験・結果

- 元の順序でFinetuningしたモデルに対して、元の順序による生成結果(Def-Def) > シャッフルした順序による生成結果(Def-Shuf)
 - **T5はデータセットの人為的整列を学習している**
- 元の順序によるFinetuning/生成 (Def-Def) v.s. シャッフルした順序によるFinetuning/生成 (Shuf-Shuf)
 - 既知ドメイン： Shuf-Shuf > Def-Def
 - 未知ドメイン： Shuf-Shuf > Def-Def (T5-Large), Def-Def > Shuf-Shuf (T5-Small, Base)
 - **シャッフルした順序でのFinetuningは、大きなモデルサイズで効果的（データ拡張的效果？）、性能向上は学習データのレコード数分布にも依存**

今後の予定

適切なプランニング手法の設計

シャッフルによる影響の解析

- エポックごとにシャッフルすればどうなるか？
- データ拡張の視点から更に検証