大規模言語モデルに対する対訳データを用いた継続事前訓練による翻訳精度評価

〇近藤 海夏斗[1]　宇津呂 武仁[1]　森下 睦[2]　永田 昌明[2]

1. 筑波大学大学院 システム情報工学研究群 自然言語処理研究室
2. NTTコミュニケーション科学基礎研究所

- **LLMと対訳データを用いた2段階の訓練を提案**
  - 提案手法
    - ① 原言語文と目的語文が交互に出現するデータで継続事前訓練
    - ② 少量の高品質な対訳データでSupervised fine-tuning
  - ベースライン
    - ① 対訳データで訓練されたVanilla transformer
    - ② LLMをそのまま少量の高品質な対訳データでSupervised fine-tuning
  - 結果
    - LLMをそのままSupervised fine-tuningしても、Vanilla transformerより精度が低い
    - 一方で、提案手法を適用すると、継続事前訓練データの原言語文と目的語文の順番と同じ翻訳方向のみ、Vanilla transformerと比べて翻訳精度が有意に上回る

- 背景
- 関連研究
- 提案法
- 実験と結果
- まとめ

LLMにプロンプトと例を与えて未知タスクを推論

- **Zero-shot:** 例を全く示さずに推論

プロンプト

English: Hello. = Japanese:

⇒

出力例

こんにちは。

- **Few-shot:** 複数の例を示してから推論

プロンプト＋例

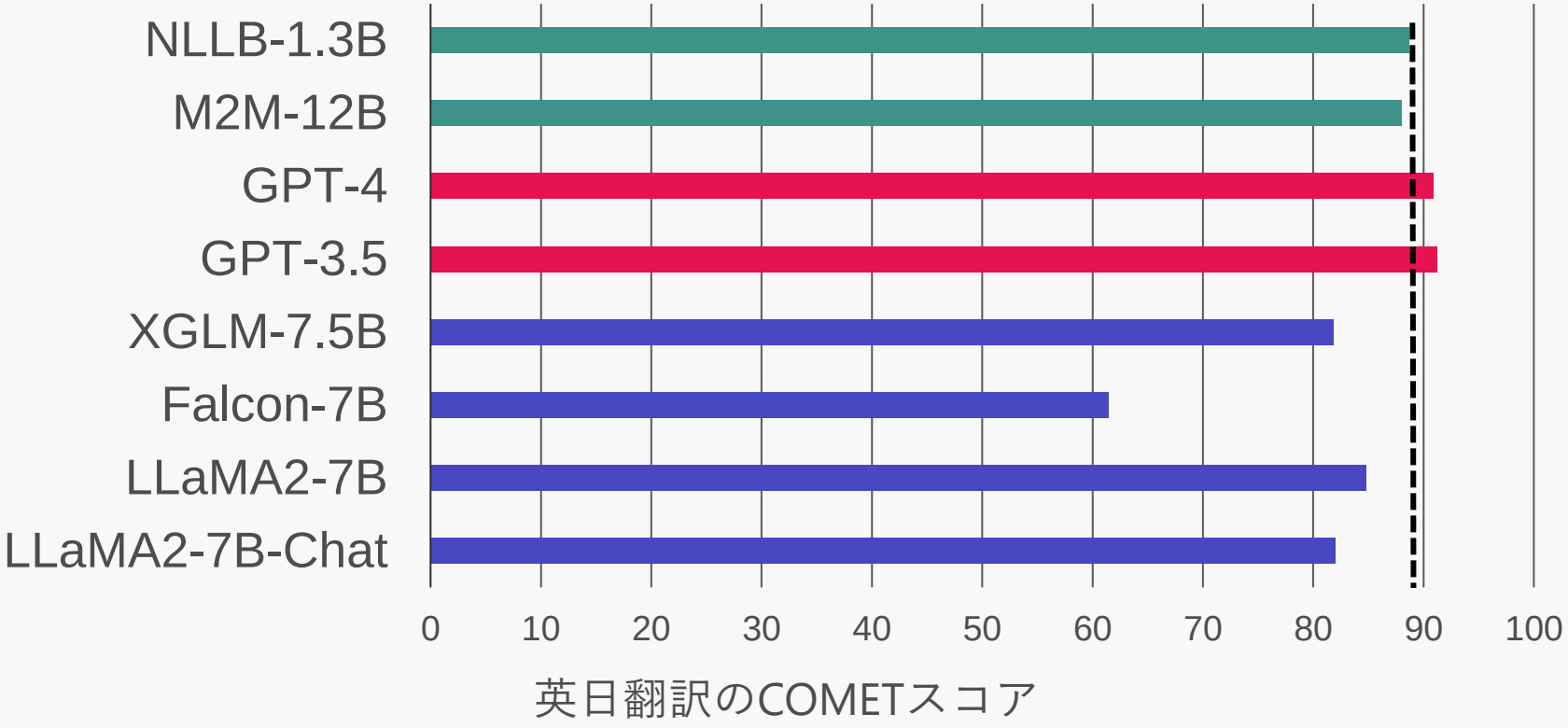
English: Thank you. = Japanese: ありがとう
English: How are you? = Japanese: 元気ですか？
English: The train is delayed. = Japanese:

⇒

出力例

電車が遅れています。

- Flores-200 [NLLB Team+, arXiv:2207.04672, 2022]のテストデータで評価
- **GPT-3.5, GPT-4のような巨大なLLM**は**既存の翻訳モデル**に匹敵
- 一方で，**7BクラスのLLM**は**既存の翻訳モデル**より大きく劣る
  - ➢ この精度の差をどうすれば縮められるか



Incidental bilingualism (翻訳例などの偶発的な二言語性)が寄与

- **事前訓練データに2言語 / 対訳データが混入**
  PaLM (540Bパラメータ)の事前訓練データ780Bトークンの中に少なくとも44言語にわたって3000万以上の対訳データが存在
- **対訳データの出現パターン**
  - Interleaved translations: {x1, y1, x2, y2}
  - Stacked translated paragraphs: {x1, x2, y1, y2}
- 1Bと8Bパラメータの LLMにおいて，**事前訓練データに2言語/対訳データを含めることで，zero-/five-shotの翻訳精度が向上**
  - 1Bと8Bでは1Bの方が精度の上昇幅が大きい

翻訳タスクだけでなく多言語タスクの精度が向上

- Encoder-Decoderモデルの事前訓練データに対訳データを含める手法を提案 [Kale+, ACL-IJCNLP2021], [Schioppa+, arXiv:2305.11778, 2023]
- **翻訳タスクに加え、QAタスク、クロスリンガル要約タスク、分類タスク、そして多言語タスク指向解析タスクの精度が向上**
- **モデルのパラメータ数が少ないほど精度が上昇する**

10B前後のLLMを2段階のfine-tuningで翻訳精度を向上

- **LLaMA2** [Touvron+, arXiv:2307.09288, 2023] (7Bと13B)
  - 事前訓練データのほとんどが英語。英語への翻訳精度はzero-shotでも高い
  - 英語以外への翻訳 (論文中では英露翻訳)では、そのまま対訳データでsupervised fine-tuningしても翻訳精度が低い
- **単言語データでfine-tuningしたのち、少量の高品質な対訳データでsupervised fine-tuning**
  - de↔en, cs↔en, is↔en, zh↔en, ru↔enの10の翻訳方向で実験
  - 1段階目の単言語データによるfine-tuningは、6つの言語の単言語データを合計1Bトークン訓練するだけでGPT-3.5のzero-shotと同等の精度を達成

事前訓練済みモデルを別データでさらに事前訓練と同じように訓練

- **LLMでは、主に英語で事前訓練されたLLaMAのようなモデルを他の言語へ転移させるために、継続事前訓練をする事例が多い**
  - Chinese-LLaMA[*1] [Cui+, arXiv:2304.08177, 2023], ELYZA-japanese-Llama-2-7b[*2], youri-7b [*3]など
- **Encoder-only、Encoder-Decoderモデルでは、任意のタスクデータを用いた継続事前訓練の有効性を報告** [Jin+, BigScience2022], [Scialom+, EMNLP2022], [Cossu+, arXiv:2205.09357, 2022]
  - LLMのようなDecoder-onlyモデルにおける有効性はまだ不透明

*1 https://huggingface.co/hfl/chinese-llama-2-7b
*2 https://huggingface.co/elyza/ELYZA-japanese-Llama-2-7b
*3 https://huggingface.co/rinna/youri-7b

- 背景
- 関連研究
- **提案法**
- 実験と結果
- まとめ

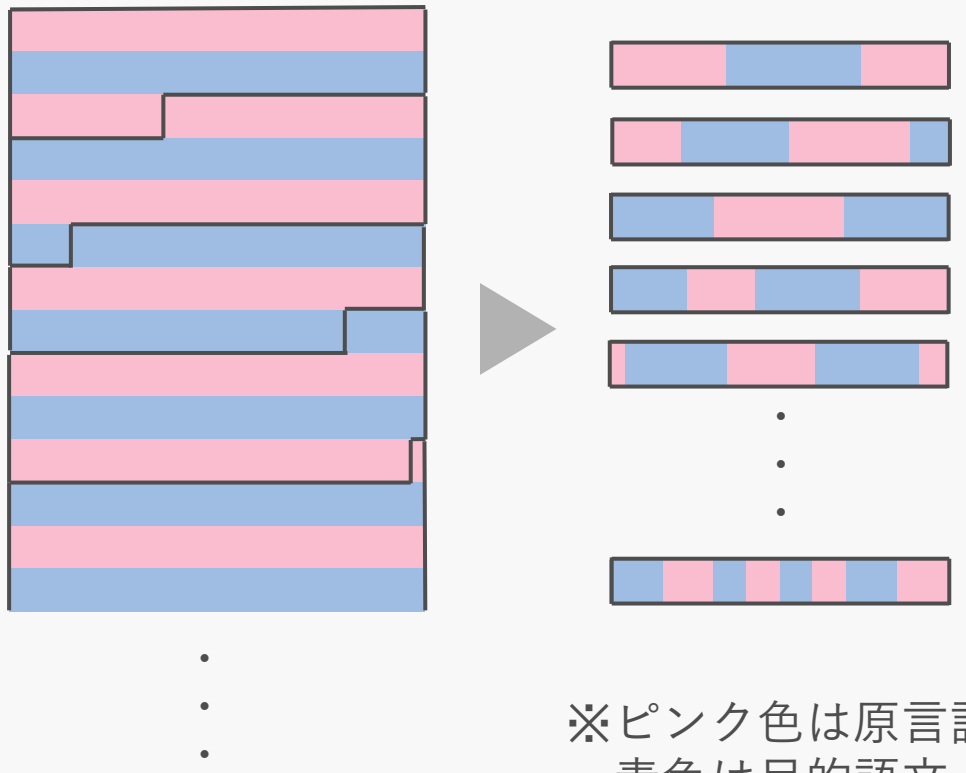
1. **原言語文と目的語文が交互に出現するデータで継続事前訓練**
   - 偶発的な二言語性を疑似的に再現するため、原言語文と目的語文を交互に結合
   - 事前訓練済みのLLMに対して，上記のデータで継続事前訓練
2. **少量の高品質な対訳データでSupervised fine-tuning**
   - 少量の高品質な対訳データは、プロの翻訳者によって作成されたデータを使用
   - 対訳データにプロンプトをつけてfine-tuning

1. **原言語文と目的語文が交互に出現するデータで継続事前訓練**
   - 偶発的な二言語性を疑似的に再現するため、原言語文と目的語文を交互に結合
   - 事前訓練済みのLLMに対して，上記のデータで継続事前訓練
2. **少量の高品質な対訳データでSupervised fine-tuning**
   - 少量の高品質な対訳データは、プロの翻訳者によって作成されたデータを使用
   - 対訳データにプロンプトをつけてfine-tuning

1. 原言語文と目的語文が交互に出現するデータで継続事前訓練

① Context length (モデルの最大入力トークン数)ずつデータを取り出す。

※Context lengthは2048, 4096などが一般的

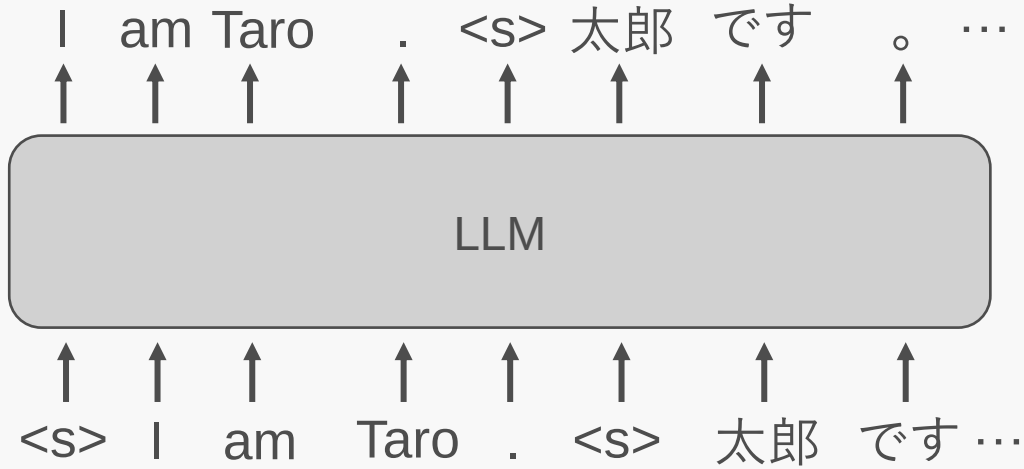


※ピンク色は原言語文
青色は目的語文

② 固定長のサンプルで各単語の次にくる単語を予測するよう訓練

入力: ①で作成した固定長のデータ

教師データ: 入力データを左に1単語だけずらす

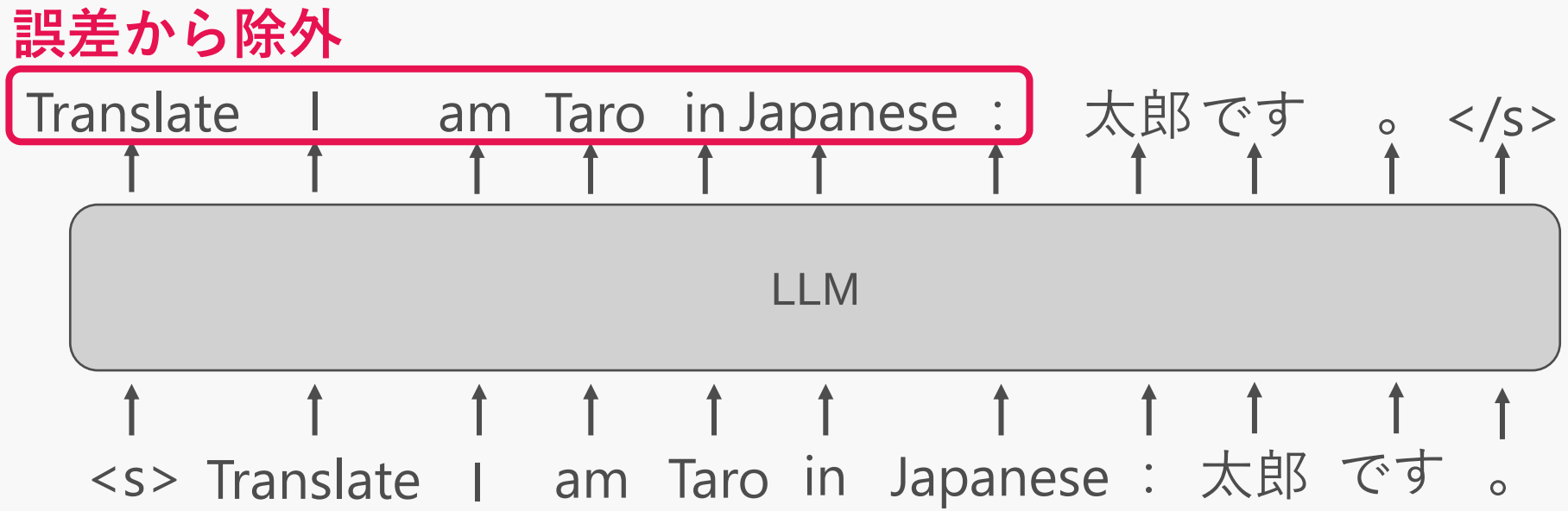


※<s>はbosトークン

1. **原言語文と目的語文が交互に出現するデータで継続事前訓練**
   - 偶発的な二言語性を疑似的に再現するため、原言語文と目的語文を交互に結合
   - 事前訓練済みのLLMに対して，上記のデータで継続事前訓練

2. **少量の高品質な対訳データでSupervised fine-tuning**
   - 少量の高品質な対訳データは、プロの翻訳者によって作成されたデータを使用
   - 対訳データにプロンプトをつけてfine-tuning

- **原言語文をもとにプロンプトを作成し、プロンプトと目的語文をそのままつなげてモデルへ入力**
- **１段階目と同様に各単語の次の単語を予測するよう訓練するが、プロンプトの出力は誤差から除外する**



※図中の<s>はbosトークン、</s>はeosトークンを表す

- **継続事前訓練**
  - **訓練データ:** 2080万文対 (JParaCrawl v3[*1]の2180万文対からLEALLA-large[*2]で抽出)
  - **開発データ:** 7990文対 (WMT20の開発とテストデータ、WMT21のテストデータ)
- **Supervised fine-tuning (SFT)**
  - **訓練データ:** 英日方向15007文対、日英方向15000文対の全30007文対
    (WMT20とFlores-200[*3]の開発・テストデータ、KFTT[*4]の訓練データ)
  - **開発データ:** 2005文対 (WMT21のテストデータ)
  - **プロンプト:** 原言語で書いたものを使用し、推論時も同じものを用いる

英日翻訳
Translate this from English to Japanese:
English: {原言語文}
Japanese:

日英翻訳
これを日本語から英語に翻訳してください:
日本語: {原言語文}
英語:

*1 [Morishita+, LREC, 2022] https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/
*2 [Mao & Nakagawa, EACL, 2023] https://huggingface.co/setu4993/LEALLA-large
*3 https://github.com/facebookresearch/flores/blob/main/flores200/README.md
*4 https://www.phontron.com/kftt/index-ja.html

13種のテストセット

翻訳方向	テストセット	分野	文数
英日・日英	ASPEC	科学技術論文	1,812
	JESC	映画字幕	2,000
	KFTT	Wikipedia記事	1,160
	TED (tst2015)	TED Talk	1,194
	Business Scene Dialogue Corpus	対話	2,120
英日のみ	WMT19 Robustness En-Ja (MTNT2019)	Reddit	1,392
	WMT20 Robustness Set1 En-Ja	Wikipediaコメント	1,100
	WMT20 Robustness Set2 En-Ja	Reddit	1,376
	IWSLT21 Simultaneous Translation En-Ja Dev	TED Talk	1,442
	WMT22 General Machine Translation Task En-Ja	ニュース、Reddit、ネット広告、対話	2,037
日英のみ	WMT19 Robustness Ja-En (MTNT2019)	Reddit	1,111
	WMT20 Robustness Set2 Ja-En	Reddit	997
	WMT22 General Machine Translation Task Ja-En	ニュース、Reddit、ネット広告、対話	2,008

- **使用するLLM: rinna/bilingual-gpt-neox-4b (rinna-4b)**[*5]
  - パラメータ数が3.8Bと比較的小規模
  - 事前訓練データは524Bトークンで英語が56% (293B)、日本語が33% (173B)
  - ➢ 事前訓練ですでに多くの単言語データで訓練されているため、ALMAの一段階目の単言語データによるfine-tuningは必要ない
- **ベースライン**
  - Transformer: JParaCrawl v3.0で訓練されたVanilla transformer[*5]
  - rinna-4b+SFT: 事前訓練済みrinna-4bをそのまま少量の高品質な対訳データでsupervised fine-tuningを行うモデル

*5 https://huggingface.co/rinna/bilingual-gpt-neox-4b, https://rinna.co.jp/news/2023/07/20230731.html
*6 推論にはhttps://github.com/MorinoseiMorizo/jparacrawl-finetuneのコードを使用

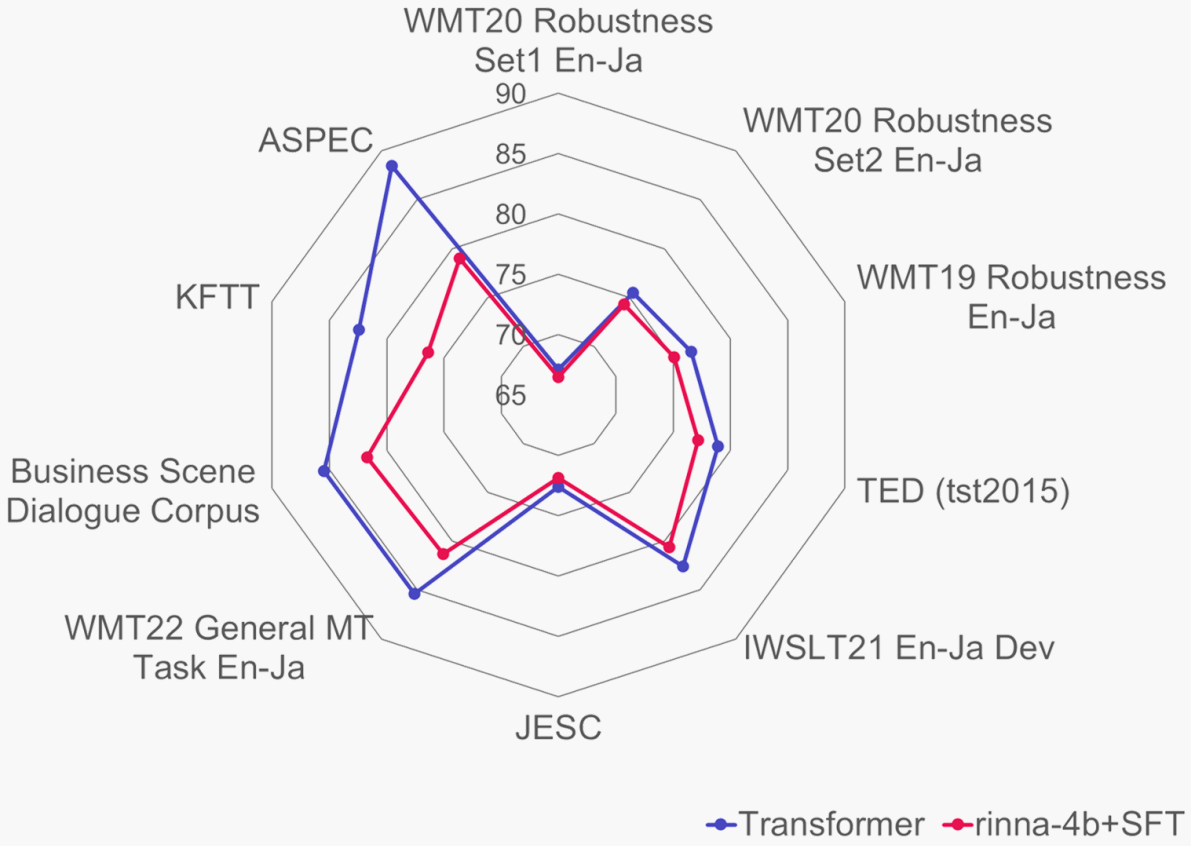
以下の4つの形式のデータで継続事前訓練

- **rinna-4b+Mono:** 対訳データを日本語と英語の単言語データとみなす
  - 英文と和文が交互になっておらず、英文もしくは和文のみのデータ
- **rinna-4b+En-Ja:** 英語から日本語の順番のみ対訳となるデータ
  - 英文の後に和訳された文章を結合する
- **rinna-4b+Ja-En:** 日本語から英語の順番のみ対訳となるデータ
  - 和文の後に英訳された文章を結合する
- **rinna-4b+Mix:** En-JaとJa-Enのデータを1040万文対ずつ重複がないようにランダムサンプリングしたデータ

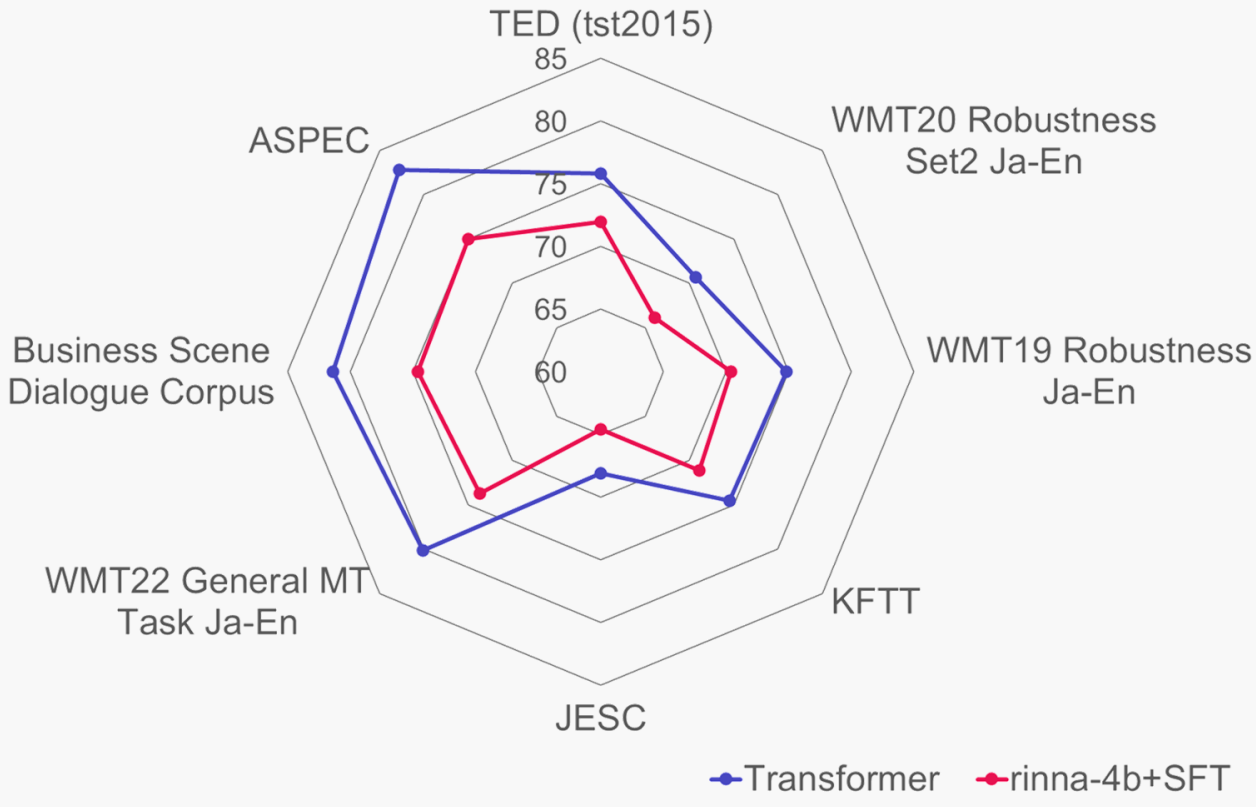
- **COMET [Rei+, WMT2022]: unbabel/wmt22-comet-daで評価*8**
  - COMETは、モデルの翻訳文と参照訳文との間で、意味がどの程度一致しているかで評価する
  - ➢ [Freitag+, WMT2022]の報告では、COMETは20個の評価指標のうち、人手評価との相関は2番目に高い (BLEUは19番目)
- **Transformerとの有意差は、有意水準5%で評価 ($p < 0.05$)**

*8 https://github.com/Unbabel/COMET

- **そのままsupervised fine-tuningしてもTransformerに精度は劣る**
  - Transformerとの有意差は現れなかった

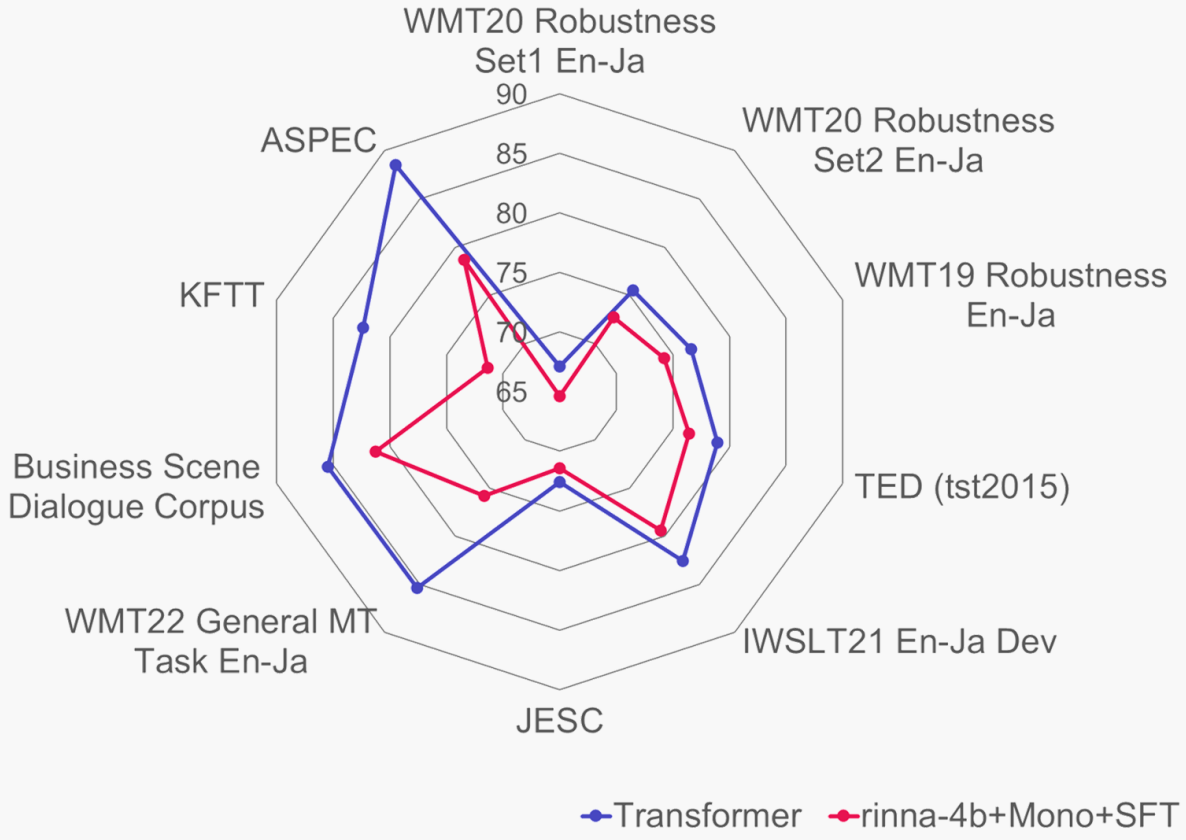


英日翻訳のCOMETスコア

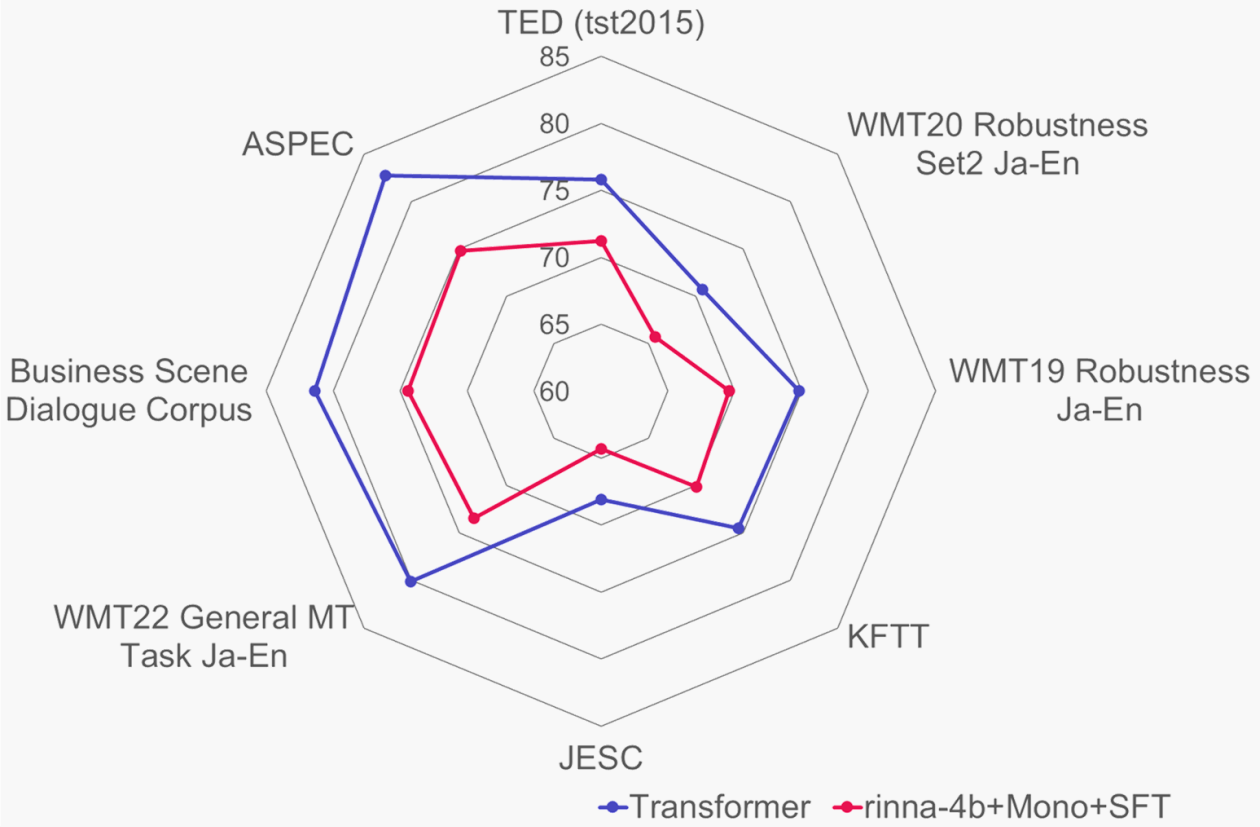


日英翻訳のCOMETスコア

- **原言語文と目的語文が交互になっていないと精度は向上しない**
  - ➢ 使用したLLMが事前訓練で既に多くの単言語データの訓練が行われているためだと考えられる

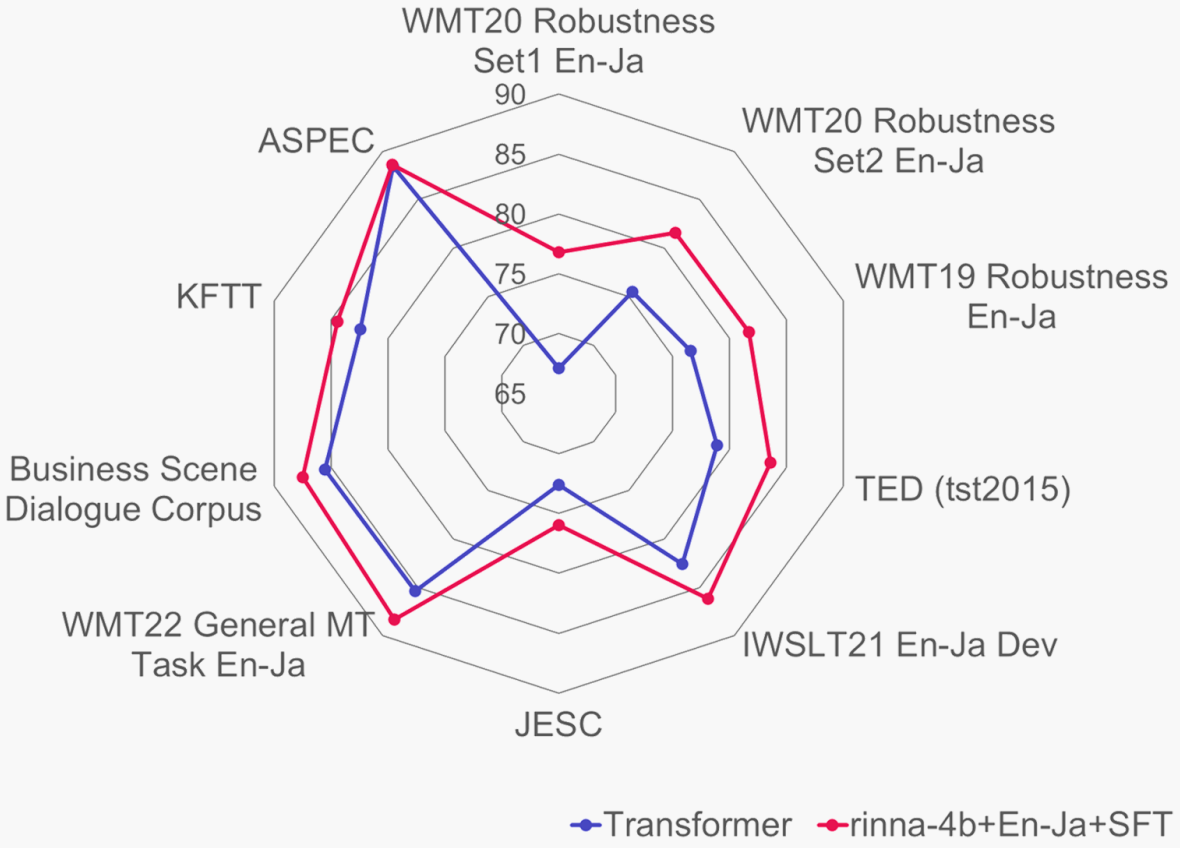


英日翻訳のCOMETスコア

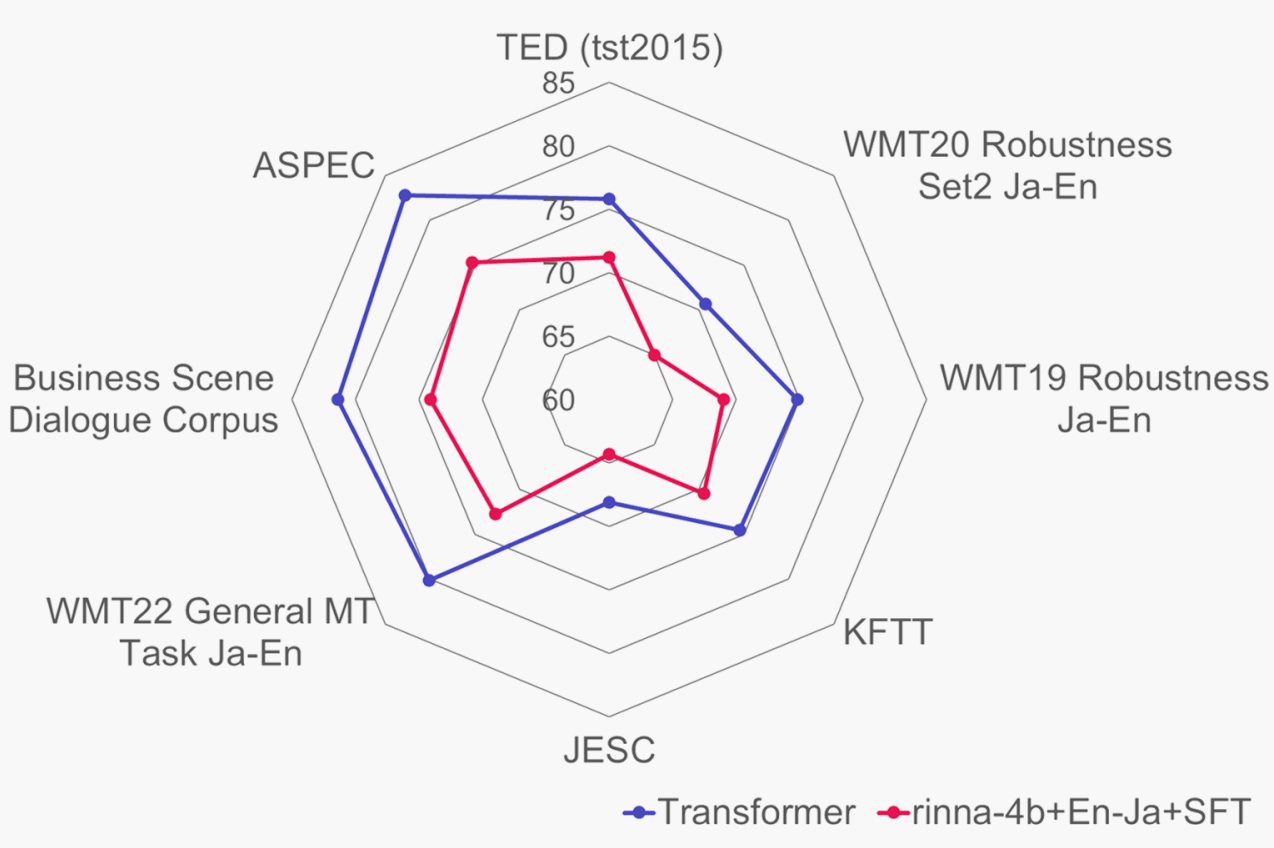


日英翻訳のCOMETスコア

- **英日翻訳の精度は向上するが、日英翻訳の精度は向上しない**
  - 英日翻訳では、ASPECを除く9種のテストセットで有意差あり

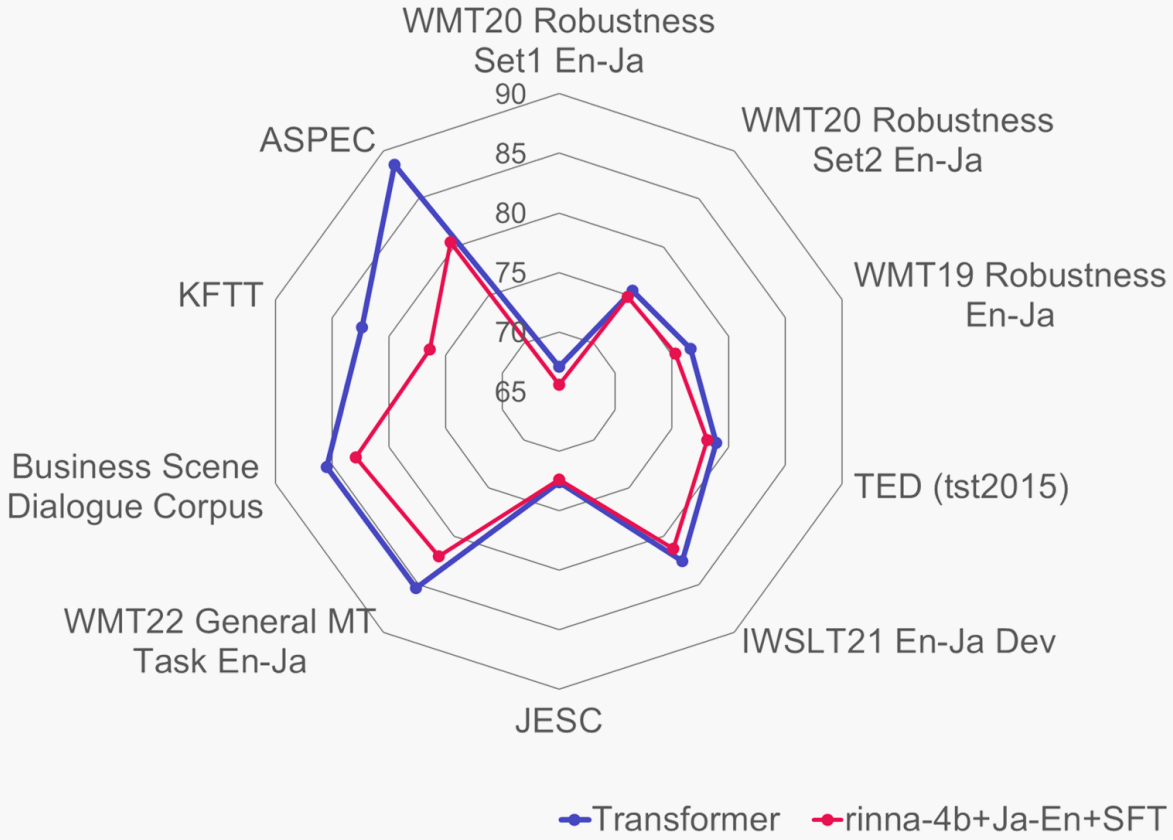


英日翻訳のCOMETスコア

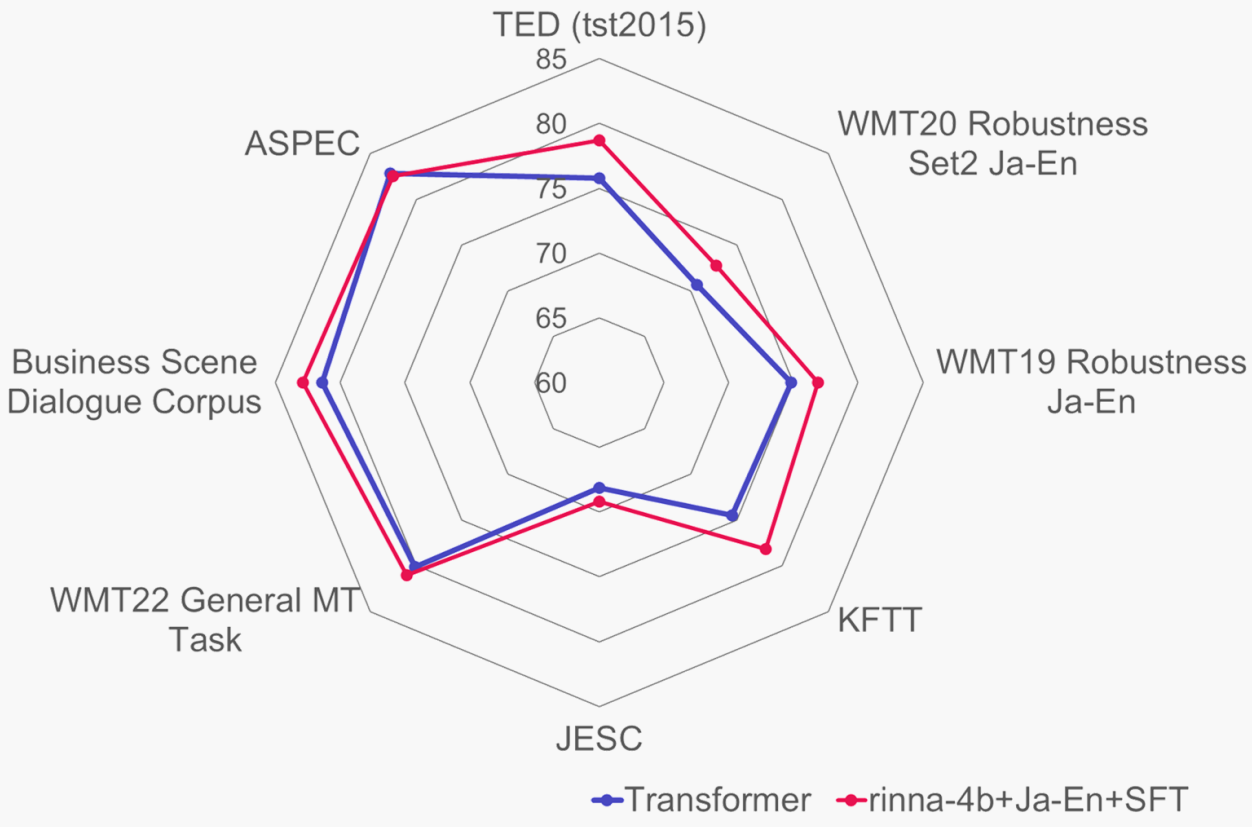


日英翻訳のCOMETスコア

- **日英翻訳の精度は向上するが、英日翻訳の精度は向上しない**
  - 日英翻訳では、ASPECを除く7種のテストセットで有意差あり

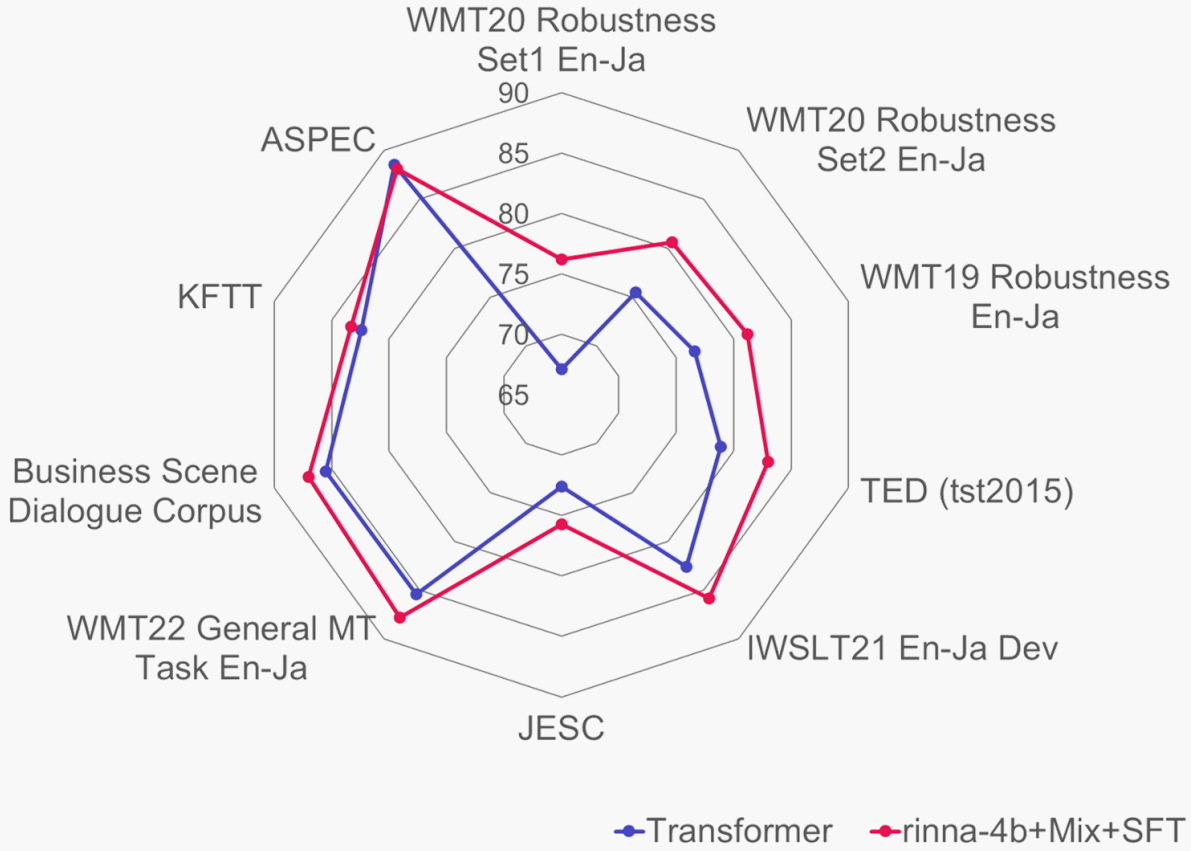


英日翻訳のCOMETスコア

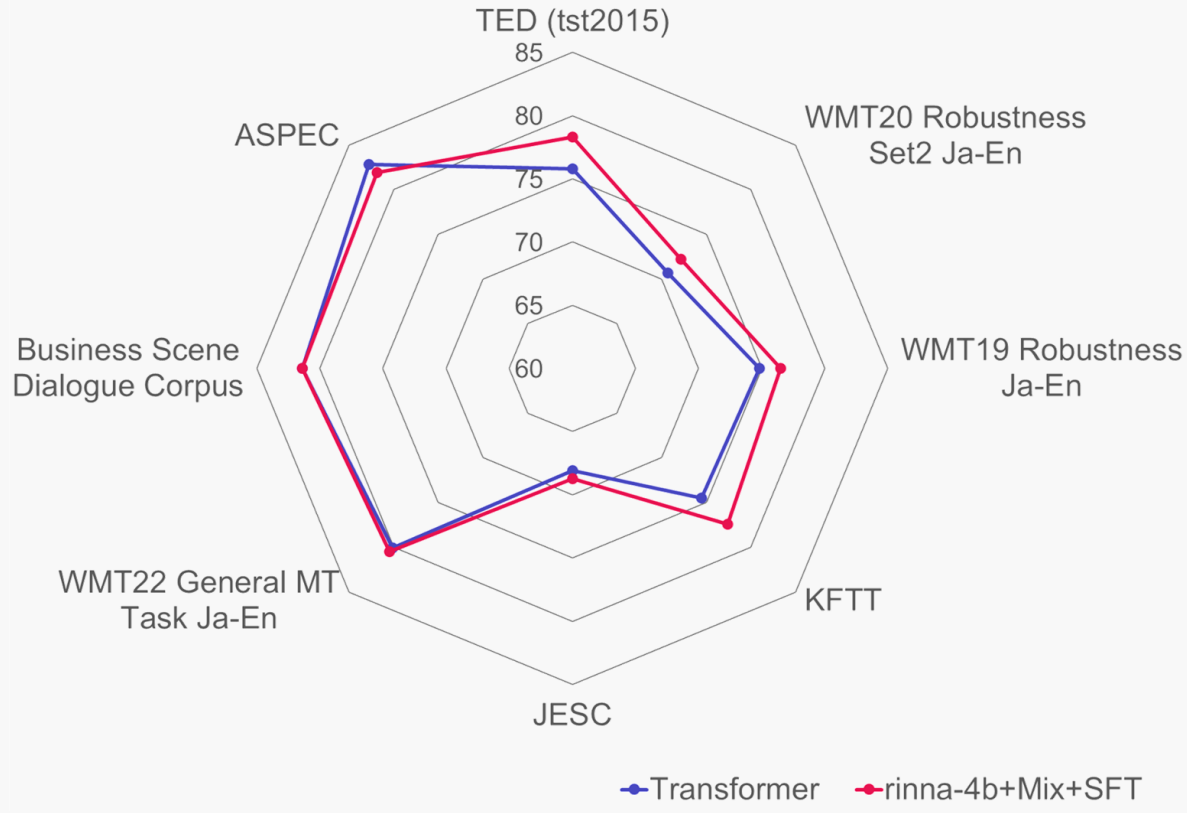


日英翻訳のCOMETスコア

- **2つの順番を両方とも訓練すると、英日翻訳では9種、日英翻訳では6種のテストセットで、Transformerと有意差あり**
  - ➢ 方向を明示しなくとも混入した対訳を見つけて活用する能力がある

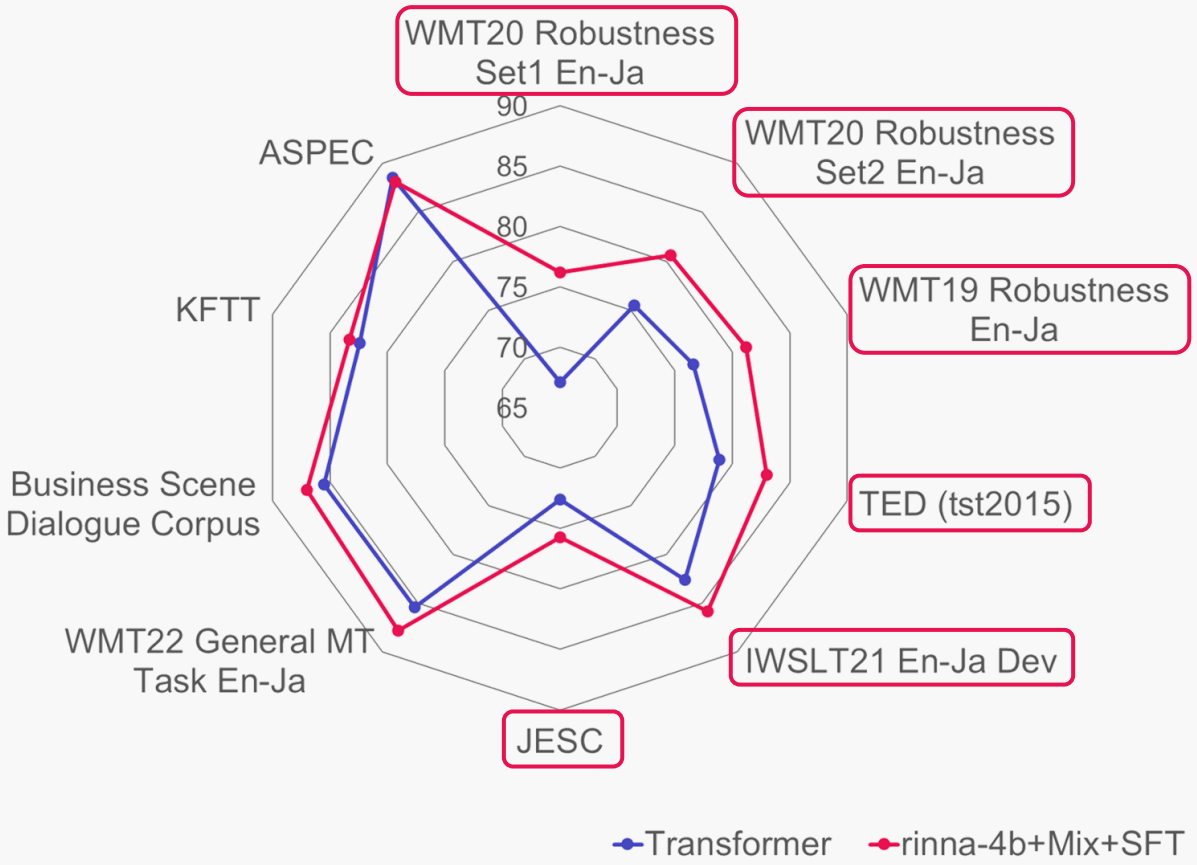


英日翻訳のCOMETスコア

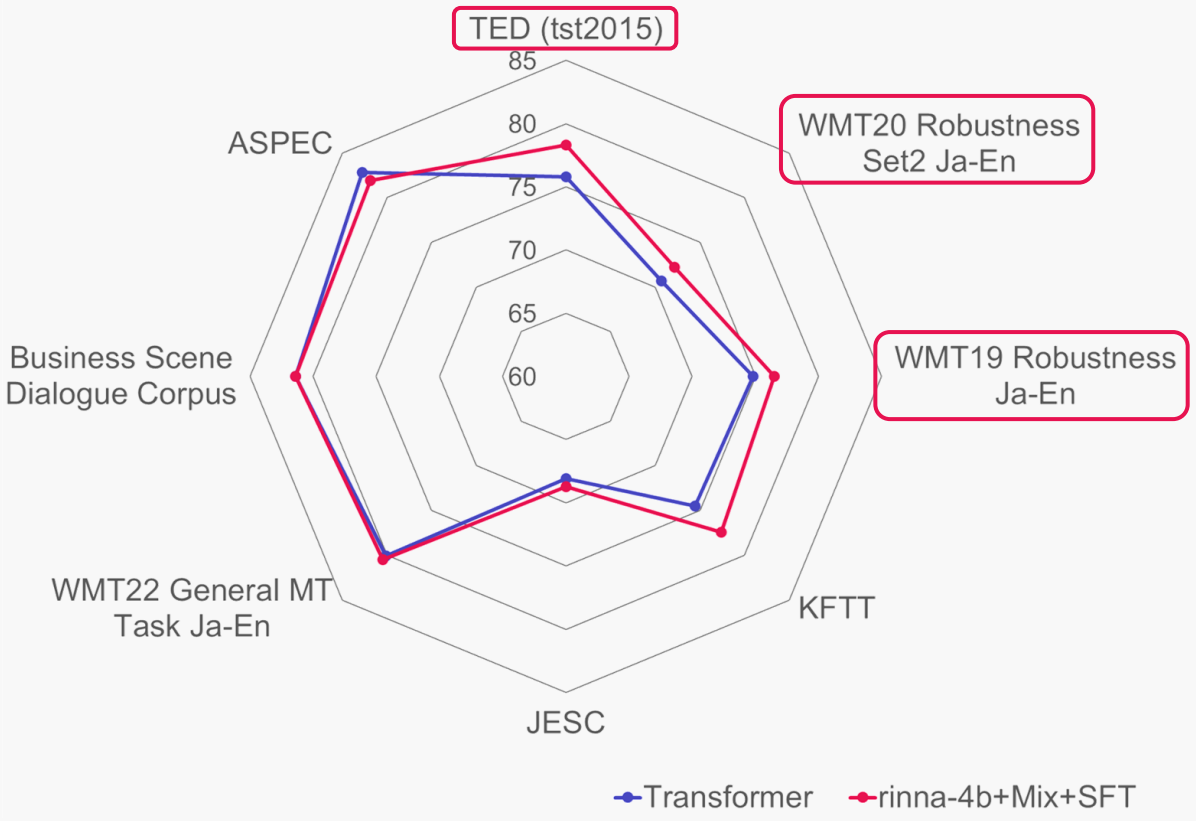


日英翻訳のCOMETスコア

- **話し言葉やノイズを多く含むデータ**が安定して高いスコア
  - ➢ **LLMは翻訳において、encoder-decoderモデルより頑健**



英日翻訳のCOMETスコア



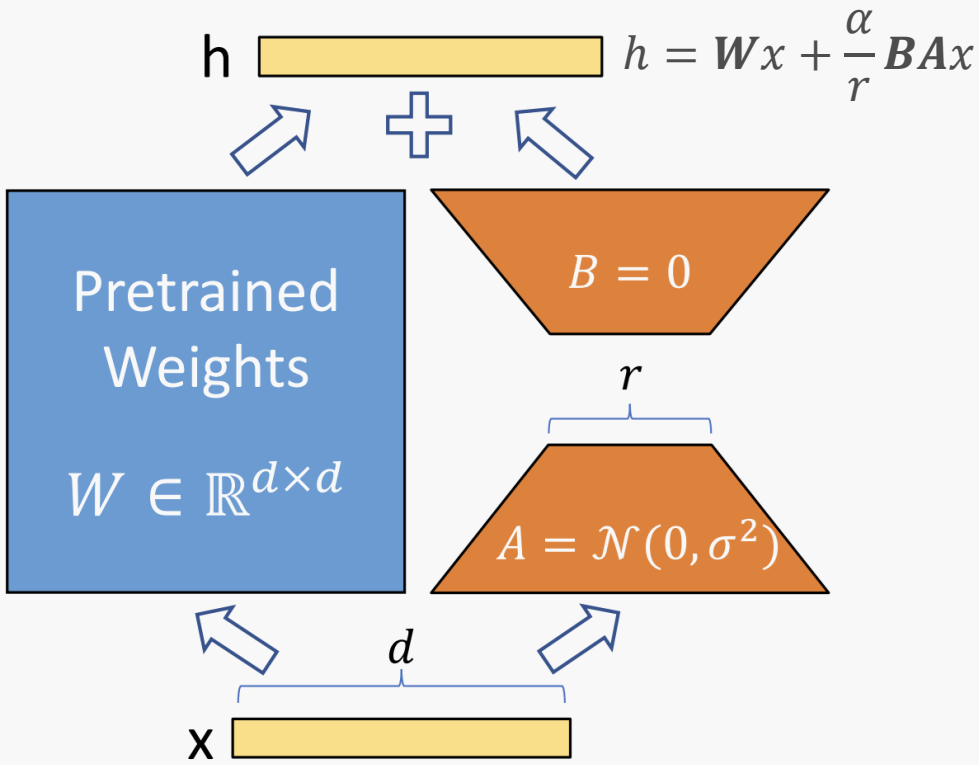
日英翻訳のCOMETスコア

- **LLMと対訳データを用いた2段階の訓練を提案**
  - 提案手法
    ① 原言語文と目的語文が交互に出現するデータで継続事前訓練
    ② 少量の高品質な対訳データでSupervised fine-tuning
  - 結果
    - LLMをそのままSupervised fine-tuningしても、Vanilla transformerより翻訳精度が低い
    - 一方で、提案手法を適用すると、継続事前訓練データの原言語文と目的語文の順番と同じ翻訳方向のみ、Vanilla transformerと比べて翻訳精度が有意に上回る
- **今後の展望**
  - 継続事前訓練を行う前と後のモデルで、QA、要約タスクなどの評価

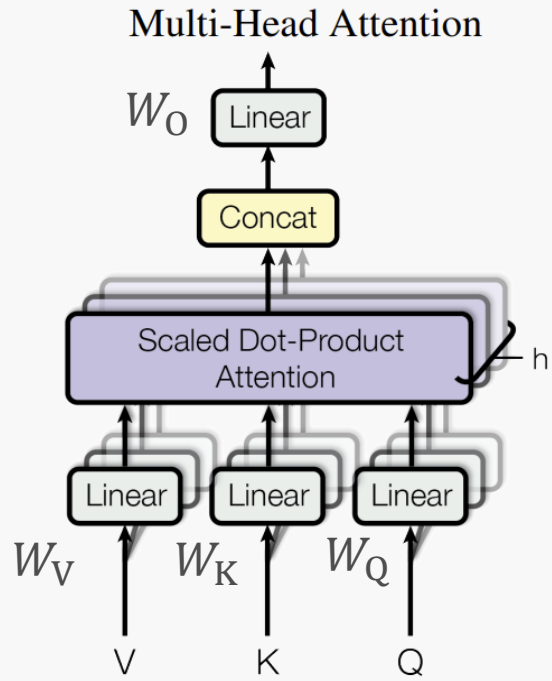
Appendix

- **LEALLA-largeから取得した文埋め込みベクトルのコサイン類似度が、0.4以上0.95未満の対訳文対をサンプリング**
  - 0.95以上は英文と和文がほとんど同じサンプルが多いため除外
  - 0.4未満は英文と和文が対訳になっていないサンプルが多いため除外
- **継続事前訓練データのトークン数**
  - rinna/bilingual-gpt-neox-4bのtokenizerで1.84B

- **r (LoRAの次元数):** 16
- **α (正規化のパラメータ):** 32
- **dropout:** 0.05



- **target_module:** Multi-Head Attentionの Query, Key, Valueの Linear (W_Q, W_K, W_V)
- **学習可能パラメータ: 6.4M (0.17%)**



※Multi-Head Attentionの図は[Vaswani+, NeurIPS, 2017]より引用

1000万文対で訓練してもTransformerを下回る

モデル	英日方向		日英方向	
	BLEU	COMET	BLEU	COMET
Transformer	21.8	85.4	21.6	80.1
rinna-4b+SFT	17.2	83.3	15.5	76.5

- **Supervised fine-tuningの訓練データ**
  LEALLA-largeから取得した文埋め込みベクトルのコサイン類似度が0.76以上0.95未満の1000万文対
- **英日方向、日英方向を別々に1epoch訓練し、WMT22のテストデータで評価**

- **継続事前訓練**
  - optimizer: AdamW (β_1 = 0.9, β_2 = 0.95, ε = 1e-8)
  - scheduler: cosine
  - weight decay: 0.1
  - gradient clipping: 1.0
  - total_batch_size: 256
  - learning rate: 1.5e-4
  - warmup ratio: 1%
  - Context length: 2048
  - bf16, deepspeed ZeRO Stage 2, gradient checkpointingを適用
  - 2台のNVIDIA RTX A6000で10日
- **Supervised fine-tuning**
  - optimizer: AdamW (β_1 = 0.9, β_2 =0.999, ε=1e-8)
  - epoch: 5
  - weight decay: 0.1
  - gradient clipping: 1.0
  - warmup ratio: 1%
  - total_batch_size: 64
  - scheduler: inverse square
  - learning rate: 3e-5(full)/2e-4(LoRA)
  - bf16, deepspeed ZeRO Stage 2, gradient checkpointingを適用

- **推論のハイパーパラメータ**
  - **Transformer**
    beam search (beam size: 6)を適用
  - **Transformer以外のモデル**
    supervised fine-tuningで開発データのlossが最小となるモデルで推論。
    bf16およびgreedy decodingを適用