大規模言語モデルによる音声対話翻訳

清水周一郎　Chenhui Chu
京都大学大学院 情報学研究科
{sshimizu,chu}@nlp.ist.i.kyoto-u.ac.jp

概要

本研究では、大規模言語モデル（LLM）を用いたcascade 方式の音声対話翻訳において、文脈情報の最適な利用方法を検討する。文脈なし・単言語文脈・二言語文脈の設定、および話者情報の有無を比較し、さらに論理的推論量が翻訳性能に与える影響を評価した。機械翻訳では文脈付与により性能が向上し、特に日英では二言語文脈が有効であることを確認した。音声翻訳では音声認識誤りによる誤差伝搬により性能が低下し、日英でその影響が大きいことが分かった。話者情報は一部条件で改善傾向が見られるものの効果は限定的であり、論理的推論量は設定により有効性が異なることが示唆された。

1　はじめに

従来の音声翻訳システムは、単言語対を想定したものが多い。しかし、異言語話者による対話において音声翻訳を用いることを考えると、二つの言語対を扱う必要がある（図 1）。この課題に対して、**音声対話翻訳**タスクが提案された [1]。

既存の音声対話翻訳研究 [1] では、mBART [2] を finetuning したモデルを二つの言語対について用意することにより、音声対話翻訳が実現されている。一方で、大規模言語モデル（LLM）の登場により、機械翻訳システムを柔軟に設計することが可能になった。LLM は一般に文脈情報を効果的に利用することで知られており、先行研究が示したように、音声対話翻訳においても文脈情報は重要である。そのため、LLM の利用による性能の向上が見込まれるが、どの程度の影響があるかは明らかでない。

加えて、最近は論理的推論[1)]を行ったのちに推論結果を出力する LLM の研究が盛んである。しかし、この LLM の論理的推論能力が音声対話翻訳に与える影響を検討した研究はない。

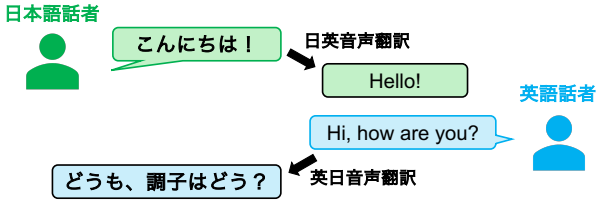

図 1　音声対話翻訳のイメージ。

さらに、音声対話翻訳では、話者情報[2)]が重要な役割を果たすと想定される。LLM の登場により、機械翻訳システムを柔軟に設計でき、話者情報の利用が容易になった。しかし、話者情報が翻訳精度に与える影響は明らかでない。

本研究では、これらの課題について、音声認識モデルと LLM を用いた cascade 音声翻訳による実験を行い、論理的推論能力や話者情報が音声対話翻訳に与える影響を、文脈情報の利用に着目しつつ検証する。

2　音声対話翻訳

2.1　音声対話翻訳タスクの概要

音声対話翻訳では、翻訳システムの仲介のもと、異なる言語を話す話者同士が対話する状況を考える。本研究では、M 人の話者 $\mathcal{S} = \{S^m \mid m = 1, 2, \ldots, M\}$ による 2 言語 $\mathcal{L} = \{L^1, L^2\}$ での対話を、音声からテキストへの翻訳システムが仲介する状況を想定する。ここで、言語 L に対してもう一方の言語を

$$\overline{L} = \begin{cases} L^1 & (L = L^2) \\ L^2 & (L = L^1) \end{cases}$$

と定義する。

対話 $D = (U_1, \ldots, U_T)$ は T 個の発話からなり、各**発話** U_t $(t = 1, \ldots, T)$ は $U_t = (S_t, L_t, X_t)$ で表される。ここで、$S_t \in \mathcal{S}, L_t \in \mathcal{L}, X_t$ はそれぞれ時点 t の発話の話者、言語、音声信号である。

1)　本稿では、英語の inference に相当する語として**推論**、reasoning に相当する語として**論理的推論**を用いる。

2)　話者情報も広義の文脈情報とみなすことができるが、本稿では文脈は対話文脈（発話内容）の意で用いる。

X_t の内容と等しい言語 L_t のテキストを $Y_t^{L_t}$ とする。すなわち、$Y_t^{L_t}$ が書き起こし、$Y_t^{\overline{L_t}}$ が翻訳テキストに相当する。

音声対話翻訳のタスクは、各発話 U_t について、音声信号 X_t から対応する翻訳テキスト $Y_t^{\overline{L_t}}$ を生成することである。

2.2 LLM による音声対話翻訳

ここでは、文脈なし、単言語文脈、二言語文脈の 3 つの設定について、話者情報を利用しない場合とする場合のそれぞれで、cascade 音声翻訳を定式化する。まずテキストの機械翻訳を定式化し、その後音声翻訳を定式化する。

簡単のため、$M = 2$ とし、話者 S^i は言語 L^i $(i = 1, 2)$ を用いるとする。また、話者は交互に発話し、話者 S^1 が対話を始めるものとする[3)]。書き起こしおよび翻訳テキストについて、正解を $Y_t^{L_t}$ や $Y_t^{\overline{L_t}}$、モデルの推論結果を $\widehat{Y_t^{L_t}}$ や $\widehat{Y_t^{\overline{L_t}}}$ で表す。

2.2.1 機械翻訳

本研究では、LLM に対するプロンプティングにより機械翻訳システムを構成する。プロンプトを構成する関数を $\phi(\cdot)$ で表すと、LLM による機械翻訳は次のように表せる。

$$\widehat{Y_t^{\overline{L_t}}} = \mathrm{Decode}\Big(p_{\mathrm{LLM}}\big(\cdot \mid \phi(Y_t^{L_t})\big)\Big)$$

ここで、Decode$(\cdot)$ はデコーディング方法を表し、例えば greedy デコーディングの場合は argmax 関数を用いて表せる。

次に、文脈を考慮した機械翻訳について考える。発話 U_t を翻訳する際の文脈を c_t とすると、

$$\widehat{Y_t^{\overline{L_t}}} = \mathrm{Decode}\Big(p_{\mathrm{LLM}}\big(\cdot \mid \phi(Y_t^{L_t}, c_t)\big)\Big)$$

である。文脈幅を c とし、文脈 c_t を

$$c_t = \big(Y_{t-c}^{\ell_{t-c}}, \ldots, Y_{t-1}^{\ell_{t-1}}\big)$$

と定義する。ここで、ℓ_{t-k} $(k = 1, 2, \cdots, c)$ は $t-k$ 番目の発話の言語であり、文脈の設定の仕方によって異なる[4)]。**単言語文脈**の場合、

$$\ell_{t-k} = L_t$$

とする。**二言語文脈**の場合、

$$\ell_{t-k} = \begin{cases} L_t & (k \bmod 2 = 0) \\ \overline{L_t}, & (k \bmod 2 = 1) \end{cases}$$

とする。

次に、話者情報を利用する場合について考える[5)]。本研究では、話者の固有名詞がテキストで与えられるものとする。文脈を考慮しない場合、

$$\widehat{Y_t^{\overline{L_t}}} = \mathrm{Decode}\Big(p_{\mathrm{LLM}}\big(\cdot \mid \phi(Y_t^{L_t}, S_t)\big)\Big)$$

であり、文脈を考慮する場合、

$$\widehat{Y_t^{\overline{L_t}}} = \mathrm{Decode}\Big(p_{\mathrm{LLM}}\big(\cdot \mid \phi(Y_t^{L_t}, S_t, c_t)\big)\Big)$$

である。文脈の構成方法は話者情報を考慮しない場合と同様である。

2.2.2 Cascade 音声翻訳

Cascade 音声翻訳は、音声認識と機械翻訳を従属接続して音声翻訳結果を得る方式である。まず、文脈を考慮しない場合を考える。音声認識モデル p_{ASR} を用いて書き起こしの推論を行う。本研究では、発話 U_t の書き起こし言語 L_t は与えられるものとし[6)]、モデルにこれを与える。

$$\widehat{Y_t^{L_t}} = \mathrm{Decode}\big(p_{\mathrm{ASR}}(\cdot \mid X_t, L_t)\big)$$

なお、音声認識における文脈の必要性は機械翻訳と比べて小さいと仮定し、ここでは文脈を考慮しない。

次に、LLM を用いた機械翻訳で翻訳の推論を行う。この際、先ほどの音声認識の推論結果を用いる。

$$\widehat{Y_t^{\overline{L_t}}} = \mathrm{Decode}\Big(p_{\mathrm{LLM}}\big(\cdot \mid \phi(\widehat{Y_t^{L_t}})\big)\Big)$$

文脈の構成方法について、基本的な定義は前節と同様であるが、文脈のどの部分が推論結果なのかが異なってくる。まず、二言語文脈については、音声認識の推論結果を全ての発話に対して適用することで構成できる。すなわち、

$$c_t = \Big(\widehat{Y_{t-c}^{\ell_{t-c}}}, \cdots, \widehat{Y_{t-1}^{\ell_{t-1}}}\Big)$$

とする。

一方で、単言語文脈については、発話 U_t の言語と同じ言語の場合には音声認識の推論結果を、それ以外の場合には cascade 音声翻訳の推論結果を用いることになる。（定式化は省略する。）

3) SpeechBSD データセットでは、同じ話者による連続した複数の発話を別の発話として扱うことがあるため、必ずしも話者が交互に発話するわけではない。また、話者が 3 者以上の場合、対話内での登場順に番号を振り、番号の偶奇が一致する話者は同じ言語を話すものとして扱う。

4) ここで定義する単言語文脈、二言語文脈では各発話で片方の言語のみを考えるが、各発話で両方の言語をどちらも考慮する方法等も考えられる。関連研究 [3] に詳しい。

5) 本研究では話者はあらかじめ与えられるものとするが、音声からの話者認識タスクを関連づけて考えることもできる。

6) 言語が与えられない設定を考えることもできる。

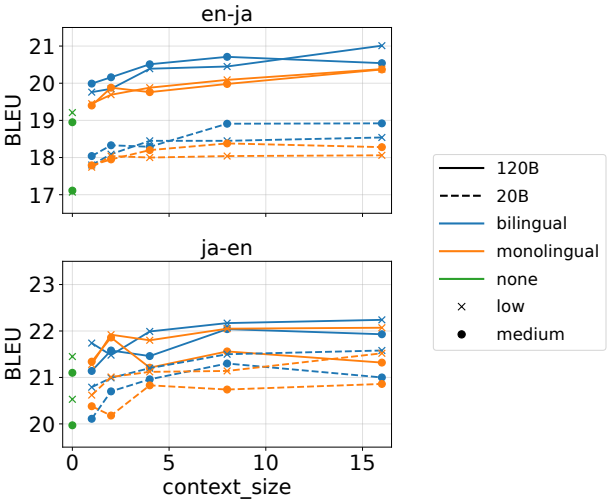

図 2　話者情報を利用しない機械翻訳の性能。横軸：文脈幅、縦軸：BLEU スコア。上：英日、下：日英。

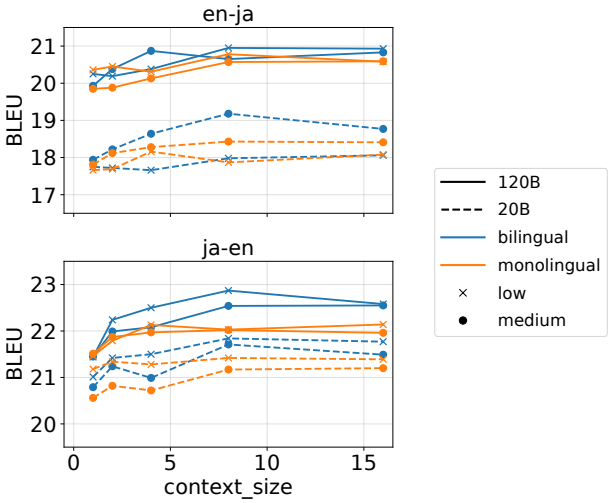

図 3　話者情報を利用した機械翻訳の性能。横軸：文脈幅、縦軸：BLEU スコア。上：英日、下：日英。

3　実験

実験には、SpeechBSD コーパス [1] の開発データを用いた。モデルの訓練などは行わず、推論のみを行った。推論は各モデルにつき 1 枚の Nvidia H100 GPU を使用し、vLLM [4] のデフォルトのハイパーパラメータでサーバを立ち上げ、機械翻訳や音声認識のクライアントがアクセスする方式とした。

3.1　機械翻訳

3.1.1　実験設定

機械翻訳を行う LLM として、OpenAI の `gpt-oss-20b` 及び `gpt-oss-120b` [5] を用いた。これらは論理的推論モデルであり、本研究では論理的推論量として `low` と `medium` で実験した[7]。最大トークン長は `low` では 1,024、`medium` では 2,048 とした[8]。文脈を与えない場合、単言語文脈の場合、二言語文脈の場合のそれぞれで異なるプロンプトを用いた。プロンプトは OpenAI harmony フォーマット[9]に従って構成した。与えたプロンプトは付録 A に示す。デコーディング方式としては、全て greedy デコーディングを用いた[10]。評価には BLEU スコア [6] を用い、SacreBLEU [7] で計算した[11]。

3.1.2　話者情報を利用しない機械翻訳

図 2 に文脈幅を変化させたときの BLEU スコアの変化を示す。文脈の与え方について、文脈を与えない場合と与える場合ではいずれも与える場合のスコアが高くなっている。単言語文脈と二言語文脈について、日英では二言語文脈の優位性が顕著に出ているのに対し、英日ではそれほど差は見られない。これは、先行研究 [1] で指摘されているように、主に代名詞の訳出において二言語文脈が有用な役割を果たしていることが原因として考えられる。また、先行研究の mBART を利用した実験では文脈幅を変えても性能はそれほど変化しなかったのに対し、本研究では顕著に差が現れたことから、LLM がより良く文脈情報を捉えられていることが示唆される。

文脈幅については、全体的に大きい方がスコアが高くなっているが、英日では安定してその傾向が見られるのに対し、日英ではあまり安定していない。モデルに関して、20B モデルより 120B モデルの方が全体的にスコアが高い。文脈幅がある程度大きくなるとスコアが停滞しているが、これは、翻訳に必要な文脈情報（代名詞の曖昧性解消のための情報など）は文脈幅を増やしてもそれほど増えないからだと考えられる。

7）予備実験として `high` も試したが、最大トークン長を 16,384 などかなり長く設定しなければならず、性能の向上に寄与する兆候も見られなかった。

8）`gpt-oss-20b` ではしばしば論理的推論のみで最大トークン長に達し推論結果が出力されないことがあり（同じトークンを繰り返す反復現象のため）、その場合は出力を空文字列とした。`gpt-oss-120b` では同様の現象は確認されなかった。

9）https://cookbook.openai.com/articles/openai-harmony

10）LLM のデコーディングでは、Softmax 関数の temperature として 0 より大きい値を用いることや、top-p や top-k サンプリングを行うことも一般的であるが、本研究では再現可能性を重視し、greedy デコーディングを用いた。

11）英日：`nrefs:1|case:mixed|eff:no|tok:ja-mecab-0.996-IPA|smooth:exp|version:2.0.0`
日英：`nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0`

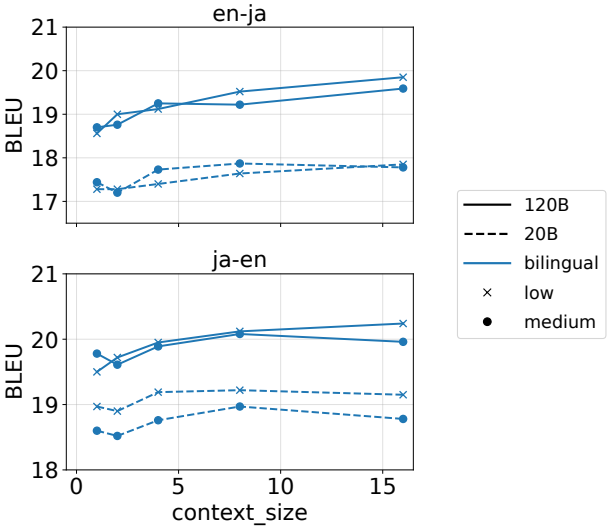

図 4　話者情報を利用しない音声翻訳の性能。横軸：文脈幅、縦軸：BLEU スコア。上：英日、下：日英。

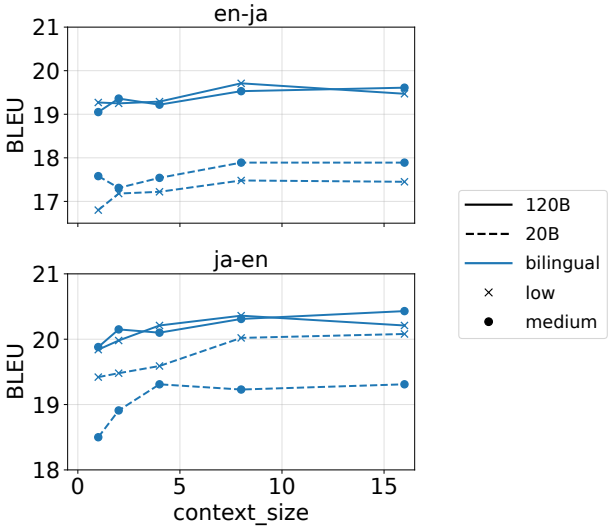

図 5　話者情報を利用した音声翻訳の性能。横軸：文脈幅、縦軸：BLEU スコア。上：英日、下：日英。

論理的推論量については、特に英日においては `medium` の方が若干 `low` よりスコアが高い傾向にあるが、日英ではあまり差が見られない。これは、論理的推論の過程が曖昧性の解消に役立っている一方で、曖昧性の解消が必要な発話が現れるのは主に英日翻訳においてであるからだと考えられる。

3.1.3　話者情報を利用した機械翻訳

話者情報を利用した機械翻訳の結果を図 3 に示す。日英の二言語文脈において、話者情報を利用しない場合と比べてスコアが良いことが確認できる。しかし、他の多くの設定ではそれほど差が見られないため、paired approximate randomization の手法による統計検定 [8] を含め、今後さらに分析を行う予定である。

3.2　音声翻訳

音声認識モデルとして、日英どちらも OpenAI の `whisper-large-v3` [9] を greedy デコーディングで用いた。機械翻訳モデルの設定は前節と同様である。ここでは、二言語文脈を利用した実験の結果のみ記す。単言語文脈の実験については今後の課題とする。

3.2.1　話者情報を利用しない音声翻訳

話者情報を利用しない音声翻訳の結果を図 4 に示す。機械翻訳と比べて全体的にスコアが低く、誤差伝搬が生じていることが分かる。特に、日英ではスコアの低下幅が大きく、音声認識モデルの日本語の性能に課題があることが分かる。本研究では LLM に音声認識結果を改善するような指示は与えていない。また、音声認識の n-best リストなども与えていないが、LLM を用いた音声からテキストへの変換タスクにおいて、n-best リストの活用は有効であることが示されている [10]。こうした情報を組み込むことで、性能が向上すると期待される。論理的推論について、特に日英のモデルパラメータ数が 20B のときに、`low` の方が `medium` よりもスコアが高い傾向が観察される。20B では不十分な情報をもとに論理的な推論を行おうとすることで却って性能が悪くなることが示唆され、今後さらなる分析を行う予定である。

3.2.2　話者情報を利用した音声翻訳

話者情報を利用した音声翻訳の結果を図 5 に示す。英日では、話者情報を利用しない場合に比べて顕著な差は見られない。日英では、特にモデルのパラメータ数が 20B のときに、話者情報を利用しない場合に比べてスコアが高くなる傾向が見られる。論理的推論量について、話者情報を利用しない音声翻訳と同様の傾向が観察される。

4　おわりに

本研究では LLM を用いた cascade 方式の音声対話翻訳において、文脈情報の利用方法に着目し、論理的推論量や話者情報が翻訳性能に与える影響を調べた。今後、単言語文脈を用いた音声対話翻訳をはじめ、本文中で言及した課題を検討する予定である。

謝辞

本研究は JSPS 科研費 JP23KJ1356 の助成を受けたものです。

参考文献

[1] Shuichiro Shimizu, Chenhui Chu, Sheng Li, and Sadao Kurohashi. Towards speech dialogue translation mediating speakers of different languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, July 2023.

[2] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 2020.

[3] Ritvik Choudhary, Rem Hida, Masaki Hamada, Hayato Futami, and Toshiyuki Sekiya. Exploring context strategies in LLMs for discourse-aware machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, November 2025.

[4] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

[5] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, July 2002.

[7] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, October 2018.

[8] Stefan Riezler and John T. Maxwell. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, June 2005.

[9] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Chiristine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision, 2022.

[10] Zhengdong Yang, Zhen Wan, Sheng Li, Chao-Han Huck Yang, and Chenhui Chu. CoVoGER: A multilingual multitask benchmark for speech-to-text generative error correction with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, November 2025.

A　プロンプト

A.1　文脈なしの場合

システムプロンプト（OpenAI harmony フォーマットにおける developer プロンプト）は以下のように設定した。

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the utterance
↪ from {src_language} to {tgt_language}. Return
↪ only the translated text without any
↪ explanations or extra text.
```

ユーザープロンプトは以下のように設定した。

```
Utterance to translate:
{utterance}
```

A.2　文脈あり、話者情報なしの場合

システムプロンプトは以下のように設定した。
単言語文脈の場合：

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the final
↪ utterance from {src_language} to
↪ {tgt_language}. Use the context utterances to
↪ resolve ambiguities. Note that the context
↪ utterances are all in {src_language}. Return
↪ only the translation of the final utterance,
↪ with no explanations.
```

二言語文脈の場合：

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the final
↪ utterance from {src_language} to
↪ {tgt_language}. Use the context utterances to
↪ resolve ambiguities. Note that the context
↪ utterances may be in either language. Return
↪ only the translation of the final utterance,
↪ with no explanations.
```

ユーザープロンプトは以下のように設定した。

```
Context utterances:
{utterance t-c}
...
{utterance t-1}

Utterance to translate:
{utterance t}
```

なお、文脈ありの設定において、各対話の最初の発話には文脈がないため、その場合は文脈なしの場合と同じプロンプトを用いた。

A.3　文脈あり、話者情報ありの場合

システムプロンプトは以下のように設定した。
単言語文脈の場合：

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the final
↪ utterance from {src_language} to
↪ {tgt_language}. Use the context utterances and
↪ speaker information to resolve ambiguities.
↪ Note that the context utterances are all in
↪ {src_language}. The format of each utterance is
↪ Speaker: Content. Return only the translation
↪ of the final utterance's content, with no
↪ explanations.
```

二言語文脈の場合：

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the final
↪ utterance from {src_language} to
↪ {tgt_language}. Use the context utterances and
↪ speaker information to resolve ambiguities.
↪ Note that the context utterances may be in
↪ either language. The format of each utterance
↪ is Speaker: Content. Return only the
↪ translation of the final utterance's content,
↪ with no explanations.
```

ユーザープロンプトは以下のように設定した。

```
Context utterances:
{speaker t-c}: {utterance t-c}
...
{speaker t-1}: {utterance t-1}

Utterance to translate:
{speaker t}: {utterance t}
```

各対話の最初の発話には以下のプロンプトを用いた。
システムプロンプト：

```
You are a machine translation system translating
↪ utterances in a dialog. Translate the utterance
↪ from {src_language} to {tgt_language}. The
↪ format of each utterance is Speaker: Content.
↪ Return only the translation of the final
↪ utterance's content, with no explanations.
```

ユーザープロンプト：

```
Utterance to translate:
{speaker}: {utterance}
```