同時通訳データを利用した 同時音声翻訳のための訳出タイミング決定手法

清水 宏晃，Graham Neubig，Sakriani Sakti，戸田 智基，中村 哲
奈良先端科学技術大学院大学 情報科学研究科
{hiroaki-sh,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

1　はじめに

音声翻訳は，ある言語を異なる言語に変換する技術であり，その性能は改善しつつある．しかし，文単位で翻訳する音声翻訳が講演のような発話が長い場面に使用される場合，一文が比較的長いため，発話開始から翻訳開始までの時間（以降，遅延時間）が長くなる問題がある．

この遅延時間の問題を解決するために，同時音声翻訳の研究が行われている [1, 2, 3]．同時音声翻訳は文単位で翻訳する音声翻訳とは異なり，文の途中で翻訳を開始するため，遅延時間を短縮できる．同時音声翻訳で重要となるのは，翻訳精度をできるだけ維持しつつ，遅延時間を短縮することである．もし，意味のまとまりを考慮せずに訳出開始を早めると，遅延時間は短縮する一方，翻訳精度は大きく劣化する [4]．特に，語順が異なる言語対においては，訳出タイミングはより重要となる．本研究の目的は，翻訳精度を維持しつつ遅延時間を短縮する訳出タイミングの考案である．

そのために我々は人間の通訳者の訳出タイミングに注目する．このことにより，巧みに同時通訳する人間の同時通訳者から，何らかのヒントが得られると考えられる．例えば，人間の同時通訳者は一文を複数に分割して翻訳する “salami technique” と呼ばれる技術を駆使することで，意味がまとまった段階で訳出を開始することが通訳に関する研究で取り上げられている [5]．そこで，本研究では同時通訳者の訳出タイミングを分析し，同時音声翻訳に利用する．

本研究ではまず，同時通訳者の同時通訳を収録したデータ (同時通訳データ) から訳出タイミングを分析し，その分析結果を基に訳出タイミング決定手法を提案する．実験では，英日方向の講演データを使用し，翻訳精度と遅延時間の観点から手法の有効性を検証する．

2　先行研究

訳出タイミングの決定手法の研究はいくつか存在する．Bangalore らは音声認識時に検出された無音区間を利用した訳出タイミング決定手法 [6] を提案した．Ryu らは翻訳時に漸次的係り受け解析を行い，構文ルールを利用した訳出タイミング決定手法 [7] を提案した．Fujita らはフレーズの並び替え確率（右確率）に基づいた多言語化に対応できる訳出タイミング決定手法 [8] を提案した．Fügen らは英西タスクにおいて，言語モデルを利用した訳出タイミング法が句読点を利用した手法と同等の性能が得られたと報告している [3]．Sridhar らも英西タスクにおいて，様々な訳出タイミング法を比較した結果，句読点と接続詞を利用した訳出タイミング法の性能が高かったと報告している [4]．

訳出タイミングの決定手法の先行研究の中には，人間の同時通訳者の技術を利用した研究は存在しない．Shimizu らによると，同時音声翻訳システムの性能は通訳経験年数 1 年の同時通訳者とほぼ同等 [1] であり，より経験年数の長い同時通訳者には未だ及んでいないと報告している．そのため，人間の同時通訳者の技術を同時音声翻訳に取り入れ，性能の改善を目指す．

3　同時通訳データの収集と分析

同時通訳者の訳出タイミングを分析するために，同時通訳データを収集した後，訳出タイミングのデータを作成し，分析する．

3.1　同時通訳データ

収録材料には TED 講演[1]を使用し，同時通訳者は講演の音声と映像を視聴しながら，英日方向へと同時通訳する．TED 講演を選んだ理由は，2 つ挙げられる．1 つ目は日本語の字幕である．TED 講演の多くには日本語の字幕が付与されており，翻訳者によって翻訳された字幕データ（翻訳データ）と同時通訳データを比較できる．2 つ目は TED 講演が機械翻訳の性能を評価する際のテストセットとして頻繁に使用されていることである．ただし，TED 講演は非常に準備され，話速が速いため，人間の同時通訳者にとって容易なタスクではない．

同時通訳者の音声を収録した後，それらのデータを書き起こした．書き起こし例を図 1 に示す．音声を 0.5 秒以上の無音区間によって分割し，タグとして時間情報（発話開始時間，発話終了時間）や言語的情報（フィラー，言いよどみ等）を付与している．また，このデータ収集に関する論文は [9] で発表する予定である．

[1] http://www.ted.com

0001 - 00:44:107 - 00:45:043
本日は<H>
0002 - 00:45:552 - 00:49:206
みなさまに(F え)難しい話題についてお話したいと思います。
0003 - 00:49:995 - 00:52:792
(F え)みなさんにとっても意外と身近な話題です。

図 1: 通訳音声の書き起こし例

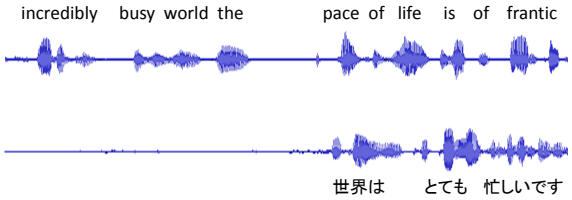

図 2: 訳出タイミングの例

3.2 訳出タイミングのデータ作成

同時通訳データから訳出タイミングを得るためには，まず話者の発話と通訳者の発話を対応づける必要がある．そのため，0.5 秒以上の無音区間で分割した通訳者の発話を発話単位と定め，その発話単位と意味が等しい話者の発話を人手によって対応づける．次に対応づけた発話単位ペアごとに，訳出タイミングを定める．その際，同時通訳者が意味のまとまった段階で同時通訳を開始することを考慮し，以下のルールで訳出タイミングを定める．

- 通訳者が訳出を開始した同時刻の話者の発話位置を訳出タイミングと定める．
- ただし，話者の発話位置の前後の単語を含む最小の句が名詞句 (NP)，前置詞句 (PP)，従属節 (SBAR) である場合は，それらの句の前を訳出タイミングとして定める．

図 2 は英語の発話音声と日本語の通訳音声の発話単位の一例を表す．同時通訳者が「世界は」と訳出したタイミングは “the” と “pace” の間である．しかし，“the” と “pace” の間は名詞句内であるため，その句の前の “world” と “the” の間を訳出タイミングと定める．句の前を訳出タイミングと定める理由は，通訳者が発話を聞いてから訳出する間にも英語の発話が流れ続けており，訳出タイミングが遅れるためである．

3.3 訳出タイミングの分析と利用法

3.2 節で作成した訳出タイミングを定めたデータ (訳出タイミングデータ) を利用して，訳出位置を分析する．具体的には，TreeTagger [10] を利用し，訳出タイミングの前後の単語に付与される品詞タグの頻度を数え上げる．データには品質が保証された通訳経験年数 15 年の同時通訳者による同時通訳データを利用し，訳出タイミングが 820 箇所存在する 7 講演分の TED データを用いた．以下に品詞タグと例を出現頻度が高い順に記述する．

表 1: 実験で用いた学習データ (train)，開発データ (dev) および テストデータ (test) に使用した TED 講演の翻訳データ (TED-T)，同時通訳データ (TED-I) および英次郎辞書と付属の例文 (DICT) の形態素数

	TED-T	TED-I	DICT
train (en)	1.68M	29.7k	13.2M
train (ja)	2.28M	33.9k	19.1M
dev (en)	–	12.1k	—
dev (ja)	–	17.4k	—
test (en)	—	13.9k	—
test (ja)	—	19.1k	—

- NN-IN (48 箇所)
  they were nothing close to the actual value⟨NN⟩ / because⟨IN⟩ almost no one ...
- NN-CC (41 箇所)
  oil is a problem⟨NN⟩ / and⟨CC⟩ coal is the most serious problem ...
- NN-PP (32 箇所)
  it is an anticorrosive heat conductor⟨NN⟩ / it⟨PP⟩ stores energy ...

訳出タイミングデータのデータサイズが小さいため，訳出タイミングデータを学習した分類器によって予測することは現実的でない．そのため，提案手法では品詞タグによる同時通訳者の訳出分析のうち，出現頻度が高かった上位 3 つの品詞タグ (NN-IN, NN-CC, NN-PP) の位置を予測して分割する．

4 実験的評価

本節では，同時通訳者の訳出タイミングを利用した手法の実験設定および実験結果について記述する．

4.1 実験設定

タスクは，TED 講演に対して英日方向の通訳である．同時音声翻訳のための機械翻訳システムの構築には [1] と同様，TED 講演の文単位の翻訳データと同時通訳データ，さらに，英次郎辞書と付属の例文 (DICT)[2]を使用する．

各データの詳細を表 1 に示す．3.3 節で同時通訳者の訳出タイミングを分析したデータは開発データに当たる．また，翻訳データと同時通訳データは統計的に異なることが示されており [1]，同時通訳者のような訳出を出力する同時音声翻訳を構築するためには，翻訳結果を翻訳データではなく，同時通訳データに対して評価せねばならない．そのため，テストの正解文は同時通訳データを使用する．

同時通訳システムは Moses [11] のフレーズベース機械翻訳を使用し，デコーディングにおける並び替えの制限 (distortion limit) は 12 と設定する．トークン

[2]http://eijiro.jp

化は英語では Moses に含まれるスクリプト，日本語では KyTea [12] を使用する．学習データ間の単語アライメントを取るツールは GIZA++ [13]，目的言語である日本語に対して言語モデルを作成するツールは SRILM [14] を使用し，5-gram で学習する．各素性の重みは MERT [15] を用いて BLEU が最大になるように最適化する．

遅延時間および翻訳精度の観点から性能を評価する．翻訳精度には，機械翻訳の自動評価尺度である BLEU [16] および RIBES [17] を用いる．評価の際，分割単位の違いにより翻訳結果と正解文の行数が異なる．そのため，翻訳精度を計算する前に，翻訳結果を翻訳精度が最大になるように分割 [18] し，正解文の行数と揃える．遅延時間は翻訳単位における発話時間と定め，発話時間は同じドメインのコーパスから学習した音響モデルを用いて推測する．原言語の 1 単語あたりの平均発話時間は開発データで 0.3073 秒，テストデータで 0.2884 秒である．本実験では，音声認識の認識性能は 100%と想定している．

4.2 比較手法

句読点に基づく手法: 句読点の位置で訳出を開始する手法であり，[3, 4] で提案されている．句読点はあらかじめ学習データに付与されたものを使用し，句読点の予測は TED の学習データを使用した LIBLINEAR [19] を用いる．単語 t_0 の後に訳出を開始するか否かは 5 つの入力単語 $t_{-2}t_{-1}t_0t_1t_2$ を基に分類し，単語 1-gram を素性に使用する．実験では特に，ピリオド (sent)，ピリオドとカンマ (punct) に基づく手法を用いる．

右確率に基づく手法: フレーズに付与された右確率によって訳出タイミングを決定する手法であり，[8] で提案されている．この手法は，あらかじめ学習データから計算された右確率のテーブルを用意しておき，そのテーブルから入力フレーズの右確率を取得してくるため，分類器は用いない．入力されたフレーズの後に訳出を開始するか否かは，設定された閾値によって決定する．実験では特に，閾値 0.80, 0.85, 0.90 とピリオドに基づく手法を組み合わせたもの (rp-0.80, rp-0.85, rp-0.90) を用いる．

品詞タグに基づく手法: 3.3 節で記述した人間の同時通訳者の訳出タイミングに基づく手法であり，本研究の提案手法である．訳出タイミングの予測は，TED の学習データを使用した LIBLINEAR [19] を用いる．単語 t_0 の後に分割できるか否かは句読点に基づく分割と同様，5 つの入力単語 $t_{-2}t_{-1}t_0t_1t_2$ を基に分類し，単語 1-gram を素性に使用する．実験では特に，3 つの手法 (NN-IN, NN-CC, NN-PP) を互いに組み合わせる．

4.3 実験結果

図 3 はテストデータにおける翻訳精度と遅延時間を示した実験結果である．従来法の句読点に基づく分割法については，punct が BLEU, RIBES 共に sent と同等の翻訳精度を維持しつつ，遅延時間を約 4 秒短縮していることが分かる．つまり，先行研究の英西タスクと同様に，英日タスクのような語順が大きく異なる言語対でも，punct は高い翻訳精度を維持しつつ遅延時間を短縮する結果になった．もう 1 つの従来法である右確率に基づく分割法については，閾値が大きくなるにつれ，翻訳精度が同等あるいは改善している一方，遅延時間も長くなることが分かる．提案手法の品詞タグに基づく分割法については，全ての品詞を組み合わせた手法 (NN-IN/CC/PP) が BLEU, RIBES 共に punct と同等の翻訳精度を得られたことが分かる．また，NN-IN/NN/CC に sent, punct を追加した手法 (NN-IN/CC/PP+sent, NN-IN/CC/PP+punct) は，翻訳精度が punct と比較して翻訳精度，遅延時間の観点から同等の性能を得られた．さらに，NN-IN/CC/PP+punct は遅延時間の標準偏差が punct と比較して小さいため，翻訳単位によって遅延時間のばらつきが少ないという利点が挙げられる．

次に，同時通訳者の遅延時間について分析した．3.3 節で作成した同時通訳者による分割位置と sent, punct をそれぞれ組み合わせた遅延時間を計算すると，それぞれ 3.11, 2.01 秒であった．つまり，通訳経験年数 15 年の同時通訳者の遅延時間は約 3 秒以内であることが分かる．また，同時通訳者に関する本には「空白の時間を 3 秒つくると，聴衆が機械トラブルを疑い不安になる.」[20] と記述されている．さらに，[1] によると通訳経験年数 4 年と 1 年の同時通訳者の遅延時間がそれぞれ 2.76 秒，2.17 秒であると報告している．それゆえ，人間の同時通訳者は約 3 秒以内に同時通訳を開始する傾向があると推測できる．図 3 において，punct, rp-0.80, NN-IN/CC/PP+sent, NN-IN/CC/PP+punct は遅延時間が約 3 秒以内であり，遅延時間が許容できる手法であることが分かる．

5 おわりに

本研究では，翻訳精度を維持したまま遅延時間を短縮する訳出タイミングの考案を目的とし，人間の同時通訳者の訳出タイミングを利用した手法を提案した．その結果，翻訳精度と遅延時間の観点から，提案手法は従来法である句読点に基づく分割法と同等の性能が得られた．また，同時通訳者の訳出タイミングの分析から，同時通訳者は発話が開始されてから約 3 秒以内に通訳を開始することが分かった．今後は，音響的な特徴量から同時通訳者の訳出タイミングを分析したい．

6 謝辞

本研究の一部は，JSPS 科研費 24240032 の助成を受け実施したものである．

参考文献

[1] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Constructing a speech translation system using simultaneous interpretation data. In *Proc. IWSLT*, 2013.

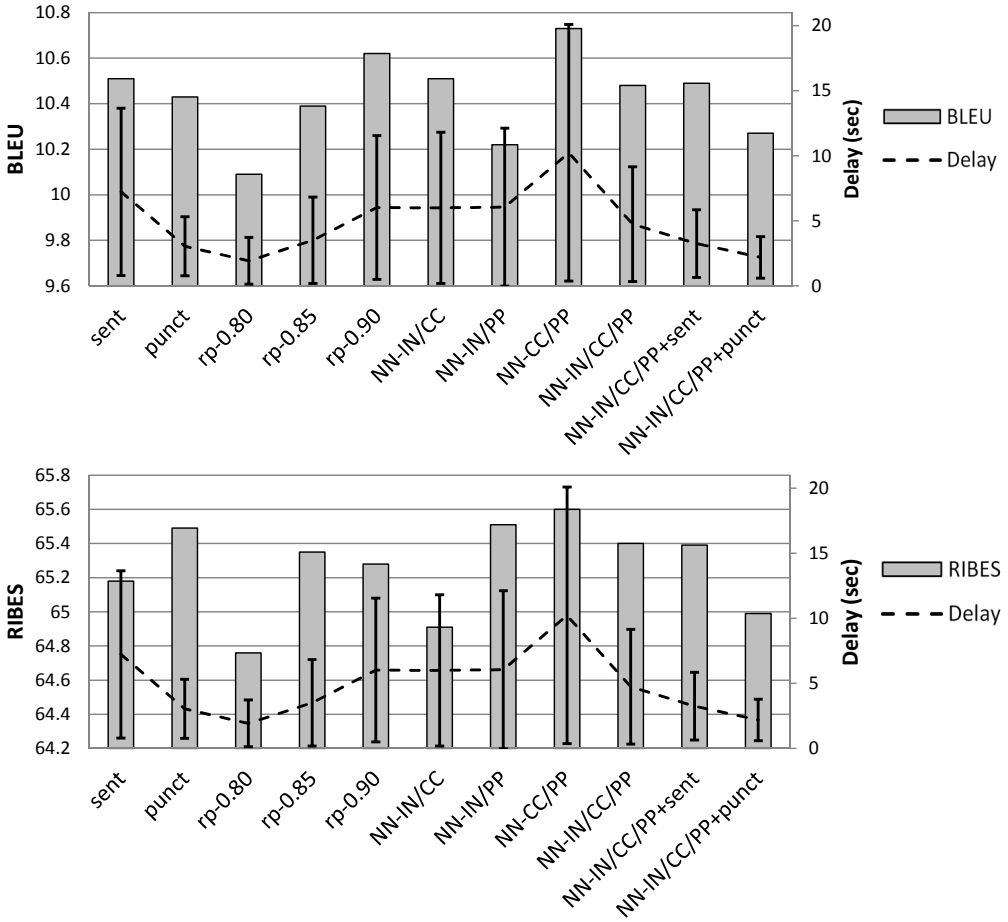

図 3: テストデータにおける翻訳結果と遅延時間

[2] Olivier Hamon, Christian Fugen, Djamel Mostefa, Victoria Arranz, Muntsin Kolss, Alex Waibel, and Khalid Choukri. End-to-end evaluation in simultaneous translation. In *Proc. EACL*, 2009.

[3] Christian Fügen, Alex Waibel, and Muntsin Kolss. Simultaneous translation of lectures and speeches. *Machine Translation*, 21(4):209–252, 2007.

[4] Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. Segmentation strategies for streaming speech translation. In *Proc. NAACL*, 2013.

[5] Roderick Jones. *Conference Interpreting Explained (Translation Practices Explained)*. St. Jerome Publishing, 2002.

[6] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan Ladan Golipour, and Aura Jimenez. Real-time incremental speech-to-speech translation of dialogs. In *Proc. NAACL*, 2012.

[7] Koichiro Ryu, Atsushi Mizuno, Shigeki Matsubara, and Yasuyoshi Inagaki. Incremental Japanese spoken language generation in simultaneous machine interpretation. In *Proc. Asian Symposium on Natural Language Processing to Overcome language Barriers*, 2004.

[8] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *Proc. InterSpeech*, 2013.

[9] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Collection of a simultaneous translation corpus for comparative analysis. In *Proc. LREC*, 2014.

[10] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees, 1994.

[11] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, 2007.

[12] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, 2011.

[13] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[14] Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Proc. ICSLP*, 2002.

[15] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, 2003.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, 2002.

[17] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, 2010.

[18] Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proc. IWSLT*, 2005.

[19] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 2008.

[20] 関谷 英里子. **同時通訳者の頭の中**. 祥伝社, 2013.