

同時機械翻訳における訳出セグメントの分割位置の推定

中林 明子

加藤 恒昭

東京大学 大学院総合文化研究科

akiyasumi@hotmail.com

kato@boz.c.u-tokyo.ac.jp

1 はじめに

同時通訳者は原文の発話を聞きながら、ほぼ同時に発話の内容を他の言語に訳出する。特に英語・日本語間など語順の異なる言語間の同時通訳においては、処理の負荷を制御し、同時性を保つために、一文を「意味の単位 (unit of meaning)」で分割し、訳出単位である訳出セグメントとし、この訳出セグメントごとに前から順に処理する。そして訳出セグメントの訳出結果を再構成、簡素化、省略することで自然な訳出を実現する [3]。同時通訳を実現する同時機械翻訳においても、同時性を保つためには、原文を訳出セグメントに分割し、それらを順次翻訳していくのが一般的な枠組みである。この枠組みのもと、本稿では同時通訳者が行っている訳出セグメントへの分割を再現することを目標に、その推定を行い、これを基に翻訳した結果を分析した。

2 背景

同時機械翻訳を実現するにあたって、翻訳を開始するタイミングを決定するために様々な研究がされてきたが、そのほとんどで、原文から翻訳を開始するタイミングを決定したり、文単位のテキスト翻訳結果を参考に翻訳を開始するタイミングを決定したりする手法が取られてきた [1, 2, 4, 7]。しかし、原文に現れるある特定の統語的特徴に基づくという仮定において決定した翻訳開始位置が、翻訳先の言語にとって最も良いタイミングであるかは明らかでない。また、テキスト翻訳文は一文単位で最適化された翻訳であるため、ここから一文より短いセグメントへ分割するための特徴を得ることは難しいと推測される。一方、同時通訳文は、同時通訳者が実際に原文を適切なタイミングで分割して翻訳を行った結果であり、訳出セグメントの分割位置を推定するにあたって、同時通訳文を用いて分割位置の特徴を得ることが望ましいと考える。

そこで本稿では、同時通訳コーパスを用いて原文を訳出セグメントに分割するモデルの構築を検討する。この基準で分割されたセグメントは、テキスト翻訳コーパスで学習した一般的な翻訳モデルによって順次翻訳され、訳出セグメントが統語的に不完全な場合はこれを再構成した上で最終的な訳出結果を出力する枠組みを想定する。

訳出セグメントへの分割によって、例えば主語の後で文が分割される場合は、名詞句からなる訳出セグメントと動詞句からなる訳出セグメントが生成される。このように統語単位が文ではない訳出セグメントの翻訳品質を高めることも同時機械翻訳における課題ではあるが、これは、上記のような一般的な翻訳モデルと再構成によって解決できると考えている。

データとして、CIAIR 同時通訳コーパス [5] を使用した。CIAIR 同時通訳コーパスには独話と対話の同時通訳データが収録されている。独話には 50 件の英語のスピーチ（発話数 2,849）を日本語に同時通訳した、136 件の同時通訳データ（発話数 5,011）が含まれる。そのうち、2000 年に収録されたデータ（24 スピーチ）は、1 スピーチにつき、4 人の通訳者が通訳を行っており、これを使用した。

3 分割位置の同定

同時通訳文を用いて、原文における訳出セグメントの分割位置を同定した。具体的には、同時通訳文における節を同時通訳者が認識した「意味の単位」であると仮定し、これに対応する原文の位置を、訳出セグメントの分割位置とした [6]。以下に例を示す。/ が「意味の単位」あるいは訳出セグメントの分割位置を示している。

通訳文：バグパイプが通常聞かれるのはハイランドゲームです。/ そこで多くのバンドが集まりまして / そして音楽を演奏します。/ そしてスコットランドのゲーム例えばケーバートスあるいはハンマースロー

などを行います。(SXZSX024.K.IA05AL)

原文: The bagpipe is most commonly heard at high-land games / where many bands gather / to play music / and perform Scottish games such as the caber toss or the hammer throw. /

1つのスピーチに対して、4人の通訳文があることから、4人のうち3人の通訳者のデータが一致した分割位置を、最終的な原文における訳出セグメントの分割位置と判断した。今回、同時通訳文における節の特定、およびこれに対応する原文の位置の特定は、人手でアノテーションを行なったが、これらの作業は自動化することも可能であると考える。

提案手法によって得た翻訳が、実際の同時通訳文に近い語順での翻訳となっていることを確認するために、上記の基準に基づいて分割したセグメントごとに google 翻訳を行い、これに対する RIBES[3] の値を、同時通訳文に対する RIBES の値と比較した。提案手法によって得た翻訳結果に対して、通訳者3人の通訳文を参照訳として RIBES の値を測定したところ、0.781253 を得た。残りの1人の通訳者の通訳文に対して、通訳者3人の通訳文を参照訳として測定して得た値は0.781565であり、一文ごとに google 翻訳を行なった結果に対して、通訳者3人の通訳文を参照訳とした場合の RIBES の値は0.748986であった。文単位の翻訳に比べ通訳者の通訳結果に大きく近づいていることから、提案手法によって同時通訳文に近い翻訳を得ることができると言える。

4 分割位置の推定

3章で述べた基準に基づいて、同時通訳文を用いて原文におけるセグメント分割位置を同定し、これを学習データとして原文から分割位置を推定する機械学習のモデルを検討した。分割位置の推定を、単語の系列を与えて、その一つ前の単語の位置で訳出セグメントが分割されるかを予測する、二値分類問題としてモデルした。

4.1 モデル

再帰的ニューラルネットワーク (RNN) の長短期記憶ユニット (Long short-term Memory, LSTM) を使用し、図1のモデルを構築した。

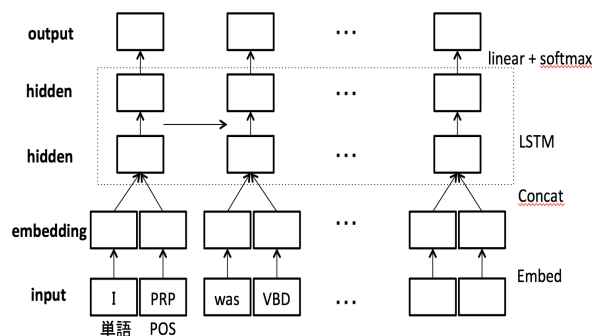


図1: セグメント分割位置を推定するモデル

入力層における単語と品詞は、ワンホットのベクトルである。これをそれぞれ、より低次元の埋め込み層に写像する。この単語と品詞の埋め込み層を結合し、これを次の LSTM の入力とする。LSTM の出力を、分類するクラスの数次元数 (今回は2次元) に持つベクトルに写像し、softmax 関数を適用することで、各クラスへの割り当ての確率を得た。損失関数としては交差エントロピーを採用した。

ニューラルネットワークの実装には Chainer¹ を使用した。ハイパーパラメータとして、埋め込み層の単語は300次元、品詞は4次元に設定した。また LSTM の入力層と出力層は埋め込み層と同じく、304次元とした。クラス別の事例数に偏りがあるため、dropout を0.1とし、また損失を計算する際に少ない事例を持つクラス1の出力の対数を3倍にするよう、重み *ClassWeight* を設定した。

なお品詞の推定には、自然言語処理ライブラリ NLTK² を用いた。NLTK では、品詞の数は、句読点などの記号を含んで45種類定義されている。

4.2 学習データと評価データ

24スピーチのうち22スピーチを学習データとして使用した。取り置いた2スピーチのうち1つを評価セットとして用い、取り置いた残りの1つは開発セットとしてパラメータを固定するために使用した。表1にデータ数の概要を示す。

データの特徴として、評価セットにおいて現れる単語のうち学習セットには現れなかった単語 (未知語) の割合が高い。また、分割位置となる単語 (クラス“1”) は全体の8.8%であり、ラベルごとのデータ数が不均衡である。

¹<https://chainer.org>

²<http://www.nltk.org>

表 1: データ概要

	学習セット	評価セット
単語数	30,151	1,197 (未知語: 231 語)
重なり語彙数	3,278	410
文の数	1,097	70
分割位置	2,662	145
分割位置以外	27,489	1,052

データに対して、以下のようにラベル付けを行い、一文単位に学習および評価を行った。“1”が割り当てられている単語の前で訳出セグメントが分割されることを示している。

単語: “I”, “was”, “traveling”, “in”, “Europe”, “and”, “when”, “I”, “was”, “in”, “Greece”, “,”, “I”, “met”, “a”, “man”, “from”, “Holland”, “.”

ラベル: (1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)

5 分割位置の推定結果

推定結果の適合率、再現率、および F 値を表 2 に示す。分割位置 (クラス “1”) の F 値として、0.79 を得た。

表 2: 訳出セグメント分割位置の推定結果

クラス	適合率	再現率	F 値
1	0.82	0.76	0.79
0	0.97	0.98	0.98

なお、4 人の通訳者間でセグメント分割位置が一致した割合が 82.5% であるため、0.825 が F 値の上限であると考えられる。

さらに、同時通訳の訳出セグメントの分割位置として特徴的に見られる、カンマ、等位接続詞 (and と or)、関係節 (which と that) における予測の結果に注目した。これらの位置ではセグメント分割が起りやすいが、必ずしも常に分割されるわけではなく、様々な要因によってその分割の有無が影響されることが観察されている [6]。カンマ、等位接続詞の直前で分割される場合は、そのほとんどは節を構成する訳出セグメントを作り出し、これは訳出セグメントの多くを占める。表 3 にそれぞれのラベル数の概要と正解率、適合率、再現率を示す。

等位接続詞 and および関係代名詞節 which においては、比較的良い適合率と再現率を得た。以下に、関係代名詞 which において、正しく分割できている事例を

表 3: 等位接続詞、カンマ、関係代名詞における推定結果

	and	or	,	which	that
ラベル数合計	1,174	264	1,382	65	205
ラベル “1” の数	776	62	632	40	36
正解率	64.0	61.7	67.9	63.1	75.1
適合率	70.8	24.0	64.1	70.0	24.1
再現率	77.4	29.0	68.0	70.0	19.4

示す。

正解: People are a little more casual, / they take their time / and they ’re really very friendly / which is something that makes me feel a lot better. / (SXUSX012)

予測: People are a little more casual, they take their time / and they ’re really very friendly / which is something that makes me feel a lot better. /

一方、関係代名詞節 that、および等位接続詞 or の適合率と再現率は低い値となった。関係代名詞節 that はその節が独立して訳出セグメントを構成する事例が少ないため、正解率は高いが、再現率が非常に低く、正しく予測できているとは言えない。また等位接続詞 or においても、適合率と再現率が低く、正しく予測できているとは言えない。

6 翻訳結果

予測した位置で訳出セグメントを分割し、セグメントごとに google 翻訳を行なった結果を分析した。本来であれば、セグメントごとに翻訳を行なった後、再構成を行なった上で最終的な翻訳結果とするが、ここでは再構成前の翻訳結果を基に分析を行った。

4 人の通訳者の通訳文を参照語とした RIBES の値は 0.775138 となり、一文ごとに google 翻訳を行なった結果に対する RIBES (0.770528) より上回る値を得た。なお、正解とした分割位置で訳出セグメントに分割し、セグメントごとに google 翻訳を行った結果に対する RIBES は 0.802530 であった。表 4 に、事例を示す。提案した手法で訳出セグメントに分割し、セグメントごとに翻訳を行うことにより、通訳例に近い語順での訳出を得ることができている。これに再構成を行うことで、さらに自然な訳出を得ることが期待できる。

予測した位置で訳出セグメントを分割した場合の翻訳結果において、低い RIBES を得た事例の一つを表 5 に示す。前置詞句の前で訳出セグメントが分割でき

表 4: 翻訳結果例

正しいセグメント位置	The bagpipe is most commonly heard at highland games / where many bands gather to play music / and perform Scottish games such as the caber toss or the hammer throw .
正しいセグメント位置に基づく翻訳結果	バグパイプはハイランドゲームでよく聞かれます たくさんのバンドが集まって音楽を演奏する そして、キャバートスやハンマースローなどのスコットランドの試合を行います。
予測したセグメント位置	The bagpipe is most commonly heard at highland games / where many bands gather to play music / and perform Scottish games such as the caber toss / or the hammer throw.
予測したセグメント位置に基づく翻訳結果	バグパイプはハイランド ゲームでよく聞かれます 音楽を演奏するために多くのバンドが集まる場所 キャバートスなどのスコットランドのゲームを実行する またはハンマー投げ
通訳例	バグパイプが通常聞かれるのはハイランドゲームです。そこで多くのバンドが集まりましてそして音楽を演奏します。そしてスコットランドのゲーム例えばケーバートスあるいはハンマースローなどを行います。

ず、また、誤った位置でセグメント分割を予測したため、RIBES の値が大きく下がる結果となった。セグメント分割の精度を改善することで、このような事例は削減できる。

7 おわりに

同時機械翻訳では、訳出単位となる訳出セグメントの決定が大きな課題の一つである。この訳出セグメントの分割位置を決定する方法として、同時通訳者の通訳文を用いて特徴を得ることを提案した。具体的には「意味の単位」は通訳文の節に現れると仮定し、これに対応する原文の位置を原文における訳出セグメントの分割位置とした。この原文における訳出セグメントの分割位置を学習データとして、ニューラルネットワークを用いて分割位置を推定する実験を行った。

分割位置における F1 値として、0.79 を得た。また、同時通訳では関係代名詞の前でセグメントを区切って訳出することがあるが、このような関係代名詞節の前に現れる分割位置を正しく推定できたケースも多く見

表 5: 最もスコアが低い文の翻訳結果

正しいセグメント位置	This happened again and again / until several people were, of course, killed.
正しいセグメント位置に基づく翻訳結果	これは何度も何度も起こった もちろん数人が殺されるまで。
予測したセグメント位置	This happened again and again until several people were, / of course, killed.
予測したセグメント位置に基づく翻訳結果	これは何人かの人々になるまで何度も何度も起こりました、もちろん、殺した。
通訳例	これが何度も何度も繰り返されました。何人かの人々が撃ち殺されるまでやったわけです。

られた。予測した位置でセグメントを分割し、セグメントごとに翻訳した結果を見てみると、同時通訳者の通訳文と近いものが多く見られ、また、それに対する RIBES は、一文ごとに翻訳した結果の RIBES より高い値を得た。しかし、現在のモデルでは、分割位置を正しく予測できていない箇所もあり、これらの位置において翻訳結果が損なわれていることから、分割位置推定の精度を高めるためのモデルの改善が今後の課題である。また、再構成を行う仕組みについても、今後検討したい。

参考文献

- [1] Grissom II, A., He, H., Boyd-Graber, J., Morgan, J., & Daume III, H. (2014). Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, 1342-1352.
- [2] Gu, J., Neubig, G., Cho, K., & Li, V. O. (2016). Learning to translate in real-time with neural machine translation. *arXiv:1610.00388*.
- [3] 平尾努, 磯崎秀樹, 須藤克仁, 塚田元, & 永田昌明. (2014). 語順の相関に基づく機械翻訳の自動評価法. *自然言語処理*, 21(3), 421-444.
- [4] Kashioka, H., Maruyama, T., & Tanaka, H. (2003). Building a parallel corpus for monologue with clause alignment. *Proceedings of the 9th Machine Translation Summit*, 216-223.
- [5] 松原茂樹, 相澤靖之, 河口信夫, 外山勝彦, & 稲垣康善 (2001). 「同時通訳コーパスの設計と構築」『通訳研究』(1), 85-102.
- [6] 中林明子, 加藤恒昭. (2018). 同時通訳のストラテジー - 訳出セグメントの分割位置とその再構成手法の分析 -. *言語処理学会第 24 回年次大会 C3-6*.
- [7] Oda, Y., Neubig, G., Sakti, S., Toda, T., & Nakamura, S. (2015). Syntax-based Simultaneous Translation through Prediction of Unseen Syntactic Constituents. *Proceedings of the 53rd ACL* (1), 198-207.