

同時音声翻訳のための 構文情報を用いた文分割に基づく機械翻訳

帖佐 克己^{1,2,a)} 小田 悠介^{2,b)} Sakriani Sakti^{2,c)} 吉野 幸一郎^{2,d)} 中村 哲^{2,e)}

概要: 同時音声翻訳システムにおいて文単位で翻訳を行うと、長文が入力された場合などに出力までの遅延が大きくなってしまいます。また、話し言葉はしばしば文同士の境界が曖昧になり、明確な文境界を仮定した翻訳手法では遅延が大きくなる可能性が高い。そこで本研究では、入力された文を短く分割しながら翻訳する方法を提案する。提案手法では、逐次的に解析される入力文の句構造情報を用い、機械学習によってリアルタイムに文の分割を行う。この結果を機械翻訳に利用することで、遅延の小さい同時音声翻訳システムを提案する。

1. はじめに

日本を訪れる外国人観光客の数が年々増加しているなど、様々な国の人々と様々な言語でコミュニケーションをする場面が増加している。また、情報通信技術の発達によって海外でのニュースや講演に触れる機会が増大しており、これを見ることで知識を深め、日常生活や業務に活かすことができる。機械翻訳はこのような場面での活用を期待されているが、スムーズに情報のやりとりを行うためには、発話に含まれている情報を正確に理解するだけでなく、内容をリアルタイムに把握する必要がある。

同時音声翻訳はリアルタイムに発話音声の話者の言語から聞き手の言語に翻訳する技術であり [1], [2], 発話内容をテキストに書き起こす音声認識、書き起こされたテキストから目的言語への翻訳を行う機械翻訳の2つから構成される。同時音声翻訳では発話中にリアルタイムで処理を行う必要があるため、機械翻訳を行う前に音声認識で得られる単語列を翻訳単位に逐次分割する必要がある。この翻訳単位のうち最も単純なものとして、文を使用し、翻訳を行う前に文境界の検出を行う手法が提案されている [4]。しかし、講義や講演などの文章では1文が長くなる傾向があり、

文の終端が来るまで機械翻訳を行わない場合、聞き手が翻訳結果を得るまでかなりの遅延が発生してしまう。これが現在の同時音声翻訳における遅延の大きな原因の1つとなっている。また、話し言葉ではしばしば文同士の境界が曖昧になることがある。この結果、複数文が結合されたものが翻訳器への入力として与えられる場合があり、明確な文境界により入力が区切られていることを仮定した手法では対処が難しい。これらの問題に対して、文という従来の処理単位ではなく文をさらに小さく分割して翻訳単位とすることが出来れば、遅延を削減できることができると考えられる。この場合、翻訳単位として適当な単位を、何らかの手法で推定することが必要となる。

同時音声翻訳システムにおける翻訳単位を決定する研究として、音声認識時に検出される無音区間によって処理単位の境界を決定する手法 [4][5] が提案されている。しかし、これらの手法は発話速度などの話者の話し方の特徴に強く影響されるという問題がある。また、現在の発話以降に入力される発話に含まれる情報を推定し分割を決定する手法もある [2]。これらに対して本研究では、逐次的句構造解析と文分割を同様の素性によって学習し、リアルタイムに文の分割を行う手法を提案するとともに、その分割結果に対する Tree-to-String 翻訳の精度を評価する。翻訳で利用する句構造の解析に用いられる情報を文分割に利用することで、適当な位置での分割が行われ、文分割による翻訳精度低下を低減することが期待できる。

2. 提案手法

本手法では、統計的機械翻訳における様々な手法の中でも Tree-to-String 翻訳を用いる。Tree-to-String 翻訳は、

¹ 大阪府立大学工業高等専門学校専攻科 総合工学システム専攻
Department of Technological Systems, Osaka Prefecture
University College of Technology Advanced Course

² 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of
Science and Technology

a) f16015@osaka-pct.ac.jp

b) oda.yusuke.on9@is.naist.jp

c) ssakti@is.naist.jp

d) koichiro@is.naist.jp

e) s-nakamura@is.naist.jp

原言語文の句構造を表す構文木を用いることで原言語文における曖昧さを低減させることが可能である。原言語と目的言語の間での文法上の関係をルールとして表現することで、これらの情報を利用しない手法よりも高い精度を実現できることが知られている。一方で、原言語の構文情報が必要となるため、漸進的な翻訳にあたっては各単語が入力された時点での構文木が必要となる。そこで今回の提案手法では、各単語の入力に合わせて逐次的に句構造解析の仮説を展開する。また、解析された句構造を Tree-to-String 翻訳の入力とする際に、どのタイミングで翻訳するかを決定する必要がある。これは最低限翻訳に必要な情報が得られた段階で翻訳を行う必要があることによる。この決定を、句構造解析を行う際と同じ素性によって実現する。本手法で用いた句構造解析の手法については 2.1 節、文の分割手法については 2.2 節で述べる。

2.1 句構造解析

句構造解析を行うには様々な方法があるが、本研究では逐次的に構文解析を行う必要があるため Shift-Reduce 構文解析を用いる。Shift-Reduce 構文解析は句構造情報を Stack と Queue 上での操作列へと変換し、分類器を用いて選んだ操作列に相当する構文木を出力する手法である。逐次的に解析ができるほか、他の手法に比べて豊富な特徴量を用いることができ、小さな計算量で比較的高い精度が得られることが知られている [3]。

Shift-Reduce 構文解析における操作は以下の 3 種類がある。

- **Shift:** Queue の先頭の要素を Stack に入れる
- **Reduce-X:** Stack の 1 番目と 2 番目を取り出して、2 つの要素の句構造関係を表す要素を Stack に挿入する
- **Unary-X:** Stack の 1 番目を取り出して、その要素の句構造関係を表す要素を Stack に挿入する

初期状態は Queue に品詞タグ付け済みの入力単語列が入っていて、Stack が空の状態とする。

本研究では、操作列を推定する分類器として構造化パーセプトロンを用いる。予測の際には、特徴量ベクトルと重みベクトルの内積をスコアとし、スコアが最も高い操作を適用する。入力ベクトルを \mathbf{x} とし、予測される出力ベクトルを \mathbf{y} 、特徴量ベクトルを $\Phi(\mathbf{x}, \mathbf{y})$ 、重みベクトルを ω とすると、予測される出力列は以下の式で表される。

$$\arg \max_{\mathbf{y}} \omega \cdot \Phi(\mathbf{x}, \mathbf{y})$$

また、予測した出力ベクトルを \mathbf{y}_{pred} 、正解の出力ベクトルを \mathbf{y}_{corr} としたとき、重みベクトルは以下の式に従って更新される。

$$\omega \leftarrow \omega - \Phi(\mathbf{x}, \mathbf{y}_{pred}) + \Phi(\mathbf{x}, \mathbf{y}_{corr})$$

2.1.1 特徴量

本研究では逐次的に解析を行うため、主辞情報を用いることができない。そこで、[6] で用いられている特徴量を拡張したものを使用する。使用した特徴量を表 1 に示す。ここで、 s_i は Stack の i 番目の要素を表す。各 s_i は素性として、直前の単語とその品詞 (**bw, bt**)、最初の単語とその品詞 (**fw, ft**)、最後の単語とその品詞 (**lw, lt**)、直後の単語とその品詞 (**aw, at**)、要素の単語長 **len**、要素の根に対応する品詞 **c** を持つ。**shape, rule** は Hall ら [7] と同様のものを用い、6 文字を超える **shape** の場合両端の 3 文字ずつを要素として用いることとした。また、 q_i は Queue の i 番目の要素を表しており、各 q_i は単語とその品詞 (**w, t**) を素性として持つ。

2.2 文の分割手法

翻訳単位を決定する文の分割にあたっては、各単語が入力されたタイミングでの句構造解析の情報を利用する。具体的には、構文解析器が Queue から Stack へ要素を移動させようとするタイミングで分類器が分割を行うか行わないかを判定する。分割を行う場合には、その時点で Queue に含まれる全ての要素に対して Shift 操作を制限した構文解析を行って 1 つの構文木を導出し、その構文木を出力することで文分割を実現する。この決定を行う分類器には、ロジスティック回帰を用いる。予測の際には、特徴量ベクトルと重みベクトルを用いて確率を計算し、確率が閾値を超えた場合は分割を行うと予測する。入力ベクトルを \mathbf{x} とし、予測される確率を $P(\mathbf{x})$ 、特徴量ベクトルを $\Phi(\mathbf{x})$ 、重みベクトルを ω とすると、予測される確率は以下の式で表される。

$$P(\mathbf{x}) = \frac{1}{1 + \exp(-\omega \cdot \Phi(\mathbf{x}))}$$

また、学習率を η 、正解の場合の確率を t としたとき、重みベクトルは以下の式に従って更新される。

$$\omega \leftarrow \omega - \eta \cdot (P(\mathbf{x}) - t) \cdot \Phi(\mathbf{x})$$

特徴量は句構造解析と同様のものを用いた。また、学習データは、データセット全体での構文木の平均単語長が目的の長さになるまで、含まれる単語の数が多い木より順番に、図 1 に示すように木構造の根から分割を行って作成した。文分割の多さ、つまり翻訳タイミングの早さと翻訳精度にはトレードオフがあることが考えられるが、一分割単位あたりの平均単語長と翻訳精度の関係については 3 節で調査する。

3. 実験

3.1 実験条件

提案した分割手法に基づく翻訳手法の精度を調査するために、英日翻訳のタスクで、さまざまな平均単語長での分

表 1 特徴量

$q_0.w \circ q_0.t$	$q_1.t \circ q_1.t$	$q_2.w \circ q_2.t$	$q_3.w \circ q_3.t$		
$s_0.c \circ s_0.ft$	$s_0.c \circ s_0.fw$	$s_0.c \circ s_0.lt$	$s_0.c \circ s_0.lw$	$s_0.c \circ s_0.at$	$s_0.c \circ s_0.aw$
$s_0.c \circ s_0.ft \circ s_0.lw$	$s_0.c \circ s_0.ft \circ s_0.lw$	$s_0.c \circ s_0.fw \circ s_0.lt$	$s_0.c \circ s_0.fw \circ s_0.lw$	$s_0.c \circ s_0.len$	$s_0.c \circ s_0.shape$
$s_0.rule$	$s_0.shape \circ s_0.rule$				
$s_1.c \circ s_1.ft$	$s_1.c \circ s_1.fw$	$s_1.c \circ s_1.lt$	$s_1.c \circ s_1.lw$	$s_1.c \circ s_1.at$	$s_1.c \circ s_1.aw$
$s_1.c \circ s_1.ft \circ s_1.lw$	$s_1.c \circ s_1.ft \circ s_1.lw$	$s_1.c \circ s_1.fw \circ s_1.lt$	$s_1.c \circ s_1.fw \circ s_1.lw$	$s_1.c \circ s_1.len$	$s_1.c \circ s_1.shape$
$s_1.rule$	$s_1.shape \circ s_1.rule$				
$s_2.c \circ s_2.ft$	$s_2.c \circ s_2.fw$	$s_2.c \circ s_2.lt$	$s_2.c \circ s_2.lw$	$s_2.c \circ s_2.at$	$s_2.c \circ s_2.aw$
$s_2.c \circ s_2.ft \circ s_2.lw$	$s_2.c \circ s_2.ft \circ s_2.lw$	$s_2.c \circ s_2.fw \circ s_2.lt$	$s_2.c \circ s_2.fw \circ s_2.lw$	$s_2.c \circ s_2.len$	$s_2.c \circ s_2.shape$
$s_2.rule$	$s_2.shape \circ s_2.rule$				
$s_0.fw \circ s_1.lw$	$s_0.ft \circ s_1.lw$	$s_0.fw \circ s_1.lt$	$s_0.ft \circ s_1.lt$	$s_0.fw \circ s_1.c$	$s_0.c \circ s_1.fw$
$s_0.lw \circ s_1.c$	$s_0.c \circ s_1.lw$	$s_1.fw \circ s_2.lw$	$s_1.ft \circ s_2.lw$	$s_1.fw \circ s_2.lt$	$s_1.ft \circ s_2.lt$
$s_1.fw \circ s_2.c$	$s_1.c \circ s_2.fw$	$s_1.lw \circ s_2.c$	$s_1.c \circ s_2.lw$	$s_0.fw \circ q_0.w$	$s_0.lw \circ q_0.w$
$s_0.fw \circ q_0.t$	$s_0.lw \circ q_0.t$	$s_0.c \circ q_0.w$	$s_0.c \circ q_0.t$	$s_1.fw \circ q_0.w$	$s_1.lw \circ q_0.w$
$s_1.fw \circ q_0.t$	$s_1.lw \circ q_0.t$	$s_1.c \circ q_0.w$	$s_1.c \circ q_0.t$	$q_0.w \circ q_1.w$	$q_0.t \circ q_1.w$
$q_0.w \circ q_1.t$	$q_0.t \circ q_1.t$				
$s_0.c \circ s_1.c \circ q_0.t$	$s_0.c \circ s_1.c \circ q_0.w$	$s_0.fw \circ s_1.c \circ q_0.t$	$s_0.lw \circ s_1.c \circ q_0.t$	$s_0.c \circ s_1.fw \circ q_0.t$	$s_0.c \circ s_1.lw \circ q_0.t$
$s_0.c \circ s_1.c \circ s_2.c$					

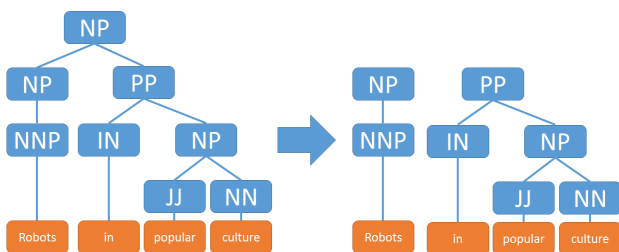


図 1 木の分割

割, および分割を行わない場合のそれぞれにおいて翻訳を行い, 自動評価尺度による比較を行った. 構文解析モデルおよび分割モデルの学習には Penn Treebank[8] の WSJ データセットを用いた. また, WIT3[9] の TED 日英音声翻訳データセットを用いて翻訳精度の評価を行った. 翻訳器の学習時には TED データに加えて, 辞書の見出し語・例文データである英辞郎データ (EIJIRO)*1, 京都フリー翻訳タスク (Kyoto)*2 のデータを用いて, 文分割を行わない場合の通常の構文解析により構文木を導出して学習を行った. 表 2 に用いたデータセットの概要を示す.

英語の品詞タグ付けには Oda によって開発された Incremental Tagger*3 (精度は約 93%) を用いる. 一般的な品詞タグ付けは文全体を用いてタグを決定する手法が多いが, 同時音声翻訳システムでは文全体を読み込んでから処理を行うと遅延が発生してしまう. そこで, 平均化パーセプトロンと過去のタグ情報を用いて逐次的に品詞タグ付け

*1 <http://eowp.alc.co.jp/info/>

*2 <http://www.phontron.com/kftt/>

*3 <https://github.com/odashi/incremental-tagger>

表 2 データセットの概要

種別	文数	単語数	
		En	Ja
WSJ-train	39.8k	950k	-
WSJ-dev	1.70k	40.1k	-
WSJ-test	2.41k	56.7k	-
TED-train	328k	2.87M	3.81M
TED-dev	8.35k	88.8k	115k
TED-test	1.76K	17.7k	23.0k
EIJIRO	2.40M	13.2M	19.1M
Kyoto	443k	11.6M	11.9M

を行うことができる Incremental Tagger を用いることにした. また, 日本語の単語分割には KyTea[10] を用いた. Tree-to-String 翻訳器は Travatar[11] を用いた.

3.2 実験結果

平均単語長と TED-test に対する翻訳結果の自動評価尺度による評価値のグラフを, それぞれ図 2 と図 3 に示す. 尺度としては, 標準的に使用される BLEU[12] および英日翻訳などの語順が大きく異なる言語対に対して有効とされる RIBES[13] を使用した. 横軸が分割手法における学習データの平均単語長, 縦軸がそれぞれの自動評価尺度の評価値を表す. また, 参考のため, 文分割を行わずに翻訳を行ったものを破線で示す (破線). この値は今回の上限値であるとともに, 最も遅延が大きくなる場合である.

実験結果より, 平均単語長が長くなるに従って翻訳精度は向上していくが, その傾きは小さくなっていくことがわかった. 特に平均単語長が 9 以降はなだらかな増加になっていることが分かる. また, 平均単語長と翻訳精度はトレードオフの関係にある. 例えば, 平均単語長が短い場合

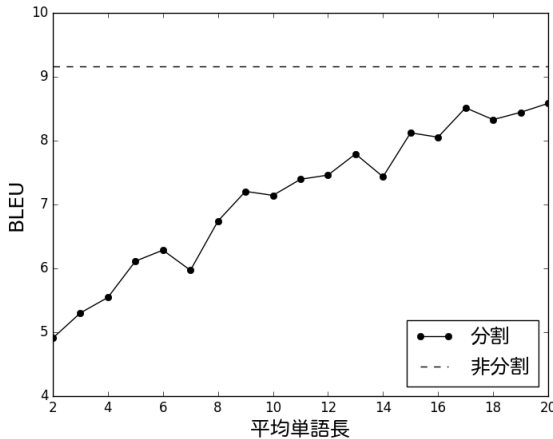


図 2 平均単語長の変化と BLEU スコア

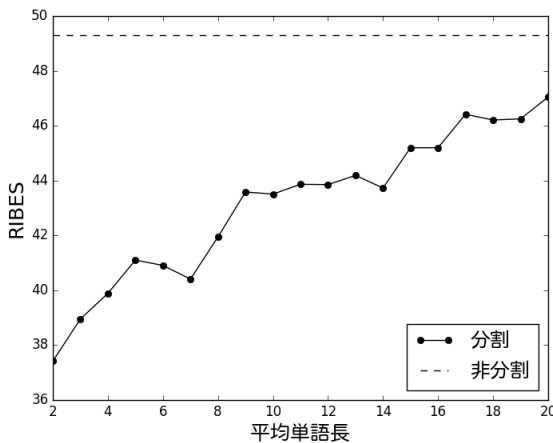


図 3 平均単語長の変化と RIBES スコア

は遅延の小さい翻訳が実現できるが、それにともなって翻訳精度は低下する。

4. おわりに

本研究では、同時音声翻訳システムのための原言語の構文情報を利用した文の分割方法を提案し、翻訳の遅延の減少を図った。この際、原言語の構文解析過程を利用して文法的構造を考慮することで、翻訳単位の構文的な妥当性を担保することを目指した。その結果、実験において10単語以下の短い平均単語長で一定の翻訳精度を実現することができた。

今後の課題としては、分割手法における分類器の学習データの作成方法の改善が挙げられる。実際の翻訳結果を考慮して、最も翻訳精度が下がらない分割を行って学習データを作成することにより、翻訳精度を下げない分割基準で学習できるのではないかと考えられる。また、現在は分割によって得られる小さい構文木を独立した単位として翻訳しているが、これらの相互作用を考慮しながら翻訳を行うことで、目的言語でより適切かつ自然性の高い翻訳が

得られるのではないかと考えられる。さらに、翻訳機そのものもあらかじめ分割されたデータによって学習することで、さらに質の高い翻訳が得られる可能性も考えられる。

参考文献

- [1] K. Kita, T. Kawabata and H. Saito. HMM continuous speech recognition using predictive LR parsing In Proc. ICASSP, 1989. ICASSP-89
- [2] Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda and Satoshi Nakamura. Syntax-based Simultaneous Translation through Prediction of Unseen Syntactic Constituents. In Proc ACL, pp198-207, 2015.
- [3] Kenji Sagae and Alon Lavie. A best-first probabilistic shift-reduce parser. In Proc. COLING/ACL, 2006.
- [4] 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 HLT, pages 437-445. 2012.
- [5] Christian Fügen, Alex Waibel, and Mumsin Kolss. Simultaneous translation of lectures and speeches. Machine Translation, Vol. 21, No. 4, pages 209-252. 2007.
- [6] Le Quang Thang, Hiroshi Noji, and Yusuke Miyao. Optimal Shift-Reduce Constituent Parsing with Structured Perceptron. In Proc. ACL, pages 1534-1544. 2015.
- [7] David Hall, Greg Durrett, and Dan Klein. Less Grammar, More Features. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 228-237, Baltimore, Maryland, June. Association for Computational Linguistics. 2014.
- [8] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The Penn Treebank. Computational linguistics, Vol. 19, No. 2, 1993.
- [9] Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit : Web inventory of transcribed and translated talks. In Proc. EAMT, pages 261-268. 2012.
- [10] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, June 2011.
- [11] Graham Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. In Proc. ACL, pages 91-96, Sofia, Bulgaria, August 2013.
- [12] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In Proc. ACL, pages 311-318, 2002.
- [13] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In Proc. EMNLP, pages 944-952, 2010.