

低リソース「日本語対応手話-日本語」対訳コーパスによる 助詞挿入と語尾活用に注目したパイプライン手話翻訳

矢野 憲
電気通信大学
yanoken@uec.ac.jp

内海 彰
電気通信大学
utsumi@uec.ac.jp

1 はじめに

手話は手指動作と被手指動作 (NMS, non-manual signals) を同時に使う視覚言語で、音声言語にならぶ固有の文法を持った言語である [1][2][3][4][5]。日本ではろう者同士、またはろう者と聴者の間で生まれ広がった日本手話 (Japanese Sign Language, JSL) のほか、日本語と手話の語をほぼ一対一に対応させた日本語対応手話、その両者の中間的な表現等が使われるが、実際には日本手話と日本語対応手話との区別は曖昧であるとされる。手話の特徴として、話言葉における自立語同士の関係性を示す付属語に相当する語が手指動作に現れない傾向がある [1][2][6]。例えば手話では質問・疑問を意味する終助詞「か」などの例外を除いて、話言葉の助詞に相当する語は表現されないことが多い。また日本語では動詞、助動詞、形容詞は語の働きの違いにより語尾が活用するが、手話では一般的にそのような語形の変化が見られないという特徴がある。このような言語的な違いに注目して日本語から手話への機械翻訳の既存研究 ([7],[8] 等) は存在するが、手話から日本語への既存研究は存在しない。従って、本研究では助詞の挿入、語尾活用に注目した日本語対応手話から日本語への機械翻訳の手法を提案した。

手話を話言葉へ翻訳するタスクでは、通常手話動画から連続する手話動作を認識し中間的な表現に変換後、その中間表現を話言葉に翻訳するという2段階で行われる [9][10][11][12]。この中間表現は一般的に *gloss* (グロス) と呼ばれる。手話を認識する問題はこれまでに *Computer Vision* の分野で活発に取り組みられてきたが、本研究では特に手話認識後の *gloss* 系列から日本語へ翻訳するタスクに取り組み、少量の手話-日本語対訳データからでも精度の高い翻訳結果を得るパイプラインによる翻訳手法を提案

した。実験の結果、提案手法の有効性を確認した。

2 対訳コーパス

2.1 日本語対応手話-日本語対訳コーパス

自動機械翻訳の目的で構築された大規模な日本手話または日本語対応手話と日本語の対訳コーパスは存在しない。このため、本研究では筆者らが現在取り組んでいる共同研究で使用している対訳コーパスを用いた。本コーパスには手話の *gloss* による中間表現と対応する日本語対訳が 1086 件存在する。本コーパスの内訳として手話の *gloss* に含まれる *gloss* 数が約 7.5K、語彙数が 655、語に分割後の日本語対訳に含まれる語数が約 11K、語彙数が 1.2K である。本コーパスの特徴として、*gloss* は gN (N は任意の数字列) のように記号表記されていることである。本コーパスでは記号表記された *gloss* と対応する日本語は別に辞書として定義している。これは、一つの *gloss* が複数の日本語へ対応しているためである。参考までに、表 1 に対訳コーパスに含まれる対訳例、表 2 に該当する *gloss* の辞書定義の例を示した。

なお、ドイツ手話 (DGS) やアメリカ手話 (ASL) の対訳コーパスでは、辞書を用いずに *gloss* が意味する話言葉における単語を大文字 (*capital letters*) で表現していることが多い [10],[6]。

表 1 手話 *gloss*-日本語対訳例

原言語 (<i>gloss</i>)	g208 g20 g28 g17 g496 g2 g84
目的言語 (日本語)	年金について相談したいです。

表 2 *gloss* 辞書定義の例

g208	g20	g28	g17	g496	g2	g84
年、 ため	お金、 費用	ついて、 ので、 だから	相談、 会議、 協議	する、 やる	欲しい、 好き、 希望	ある、 です

2.2 日本語コーパス

本研究では、手話と日本語との対訳コーパスの他に、現代日本語書き言葉均衡コーパス (BCCWJ) から生成した擬似日本語対訳コーパスを用いた。これは対訳コーパスではサンプル数が十分でないために、データ駆動による深層学習を用いた機械翻訳手法を適用することが困難なためである。本研究で用いた日本語対応手話-日本語対訳コーパスは、会話における話言葉が中心であるため、そこに現れる文末のパターン (~ か?, ~ しました, ~ ほしいです, ~ います, ~ します, ~ 希望する, 等) で BCCWJ コーパスの検索を行い該当する約 20 万の文を選択した。選択された日本語コーパスは助詞の挿入や、語の語尾活用を行う機械学習モデルの教師データとして用いた。詳細については 4. モデルの学習で述べる。

3 翻訳手法

定義

\mathcal{G} は長さ N の gloss 系列 $\mathcal{G} = (g_1, \dots, g_N)$ を示し、 \mathcal{S} は長さ U の日本語列 $\mathcal{S} = (w_1, \dots, w_U)$ を示す。 \mathcal{S} に対してさらに助詞、語尾活用を示す 2 つの添字 PP と C を定義し、それぞれの添字の前の $+$ と $-$ で助詞の挿入、語尾活用が適用されているか否かを示す。例えば \mathcal{S}_{-PP-C} は助詞がなく、語の活用形を基本形にした日本語列を示し、 \mathcal{S}_{+PP+C} は助詞の挿入、語尾活用後の日本語列を示す。

アルゴリズム

提案する gloss 系列から日本語への翻訳アルゴリズムを Algorithm 1 に以下に示した。

各 Step で行う処理を以下で説明する。

- Step 0: 統計的機械翻訳の手法を用いて gloss 系列を対応する日本語列へ変換する。
- Step 1: seq2seq 機械翻訳モデル (m1) を用いて Step 0 で得られた日本語列に対して助詞を挿入する。
- Step 2: seq2seq 機械翻訳モデル (m2) を用いて Step 1 で得られた日本語列に対して活用による語尾活用を行う。
- Step 3: Step 2 で得られた日本語列から助詞を削除する。
- Step 4: seq2seq 機械翻訳モデル (m3) を用いて Step 3 で得られた日本語列に対して助詞を挿入

Algorithm 1 gloss 系列から日本語への翻訳

Input: \mathcal{G}

Output: \mathcal{S}_{+PP+C}

Step 0: $\mathcal{G} \rightarrow \mathcal{S}_{-PP-C}$

Step 1: $\mathcal{S}_{-PP-C} \rightarrow \mathcal{S}_{+PP-C}$

Step 2: $\mathcal{S}_{+PP-C} \rightarrow \mathcal{S}_{+PP+C}$

$\mathcal{S}_{prev} = \emptyset$

$\mathcal{S}_{next} = \mathcal{S}_{+PP+C}$

while $\mathcal{S}_{prev} \neq \mathcal{S}_{next}$ **do**

$\mathcal{S}_{prev} = \mathcal{S}_{next}$

 Step 3: $\mathcal{S}_{+PP+C} \rightarrow \mathcal{S}_{-PP+C}$

 Step 4: $\mathcal{S}_{-PP+C} \rightarrow \mathcal{S}_{+PP+C}$

 Step 5: $\mathcal{S}_{+PP+C} \rightarrow \mathcal{S}_{+PP-C}$

 Step 2: $\mathcal{S}_{+PP-C} \rightarrow \mathcal{S}_{+PP+C}$

$\mathcal{S}_{next} = \mathcal{S}_{+PP+C}$

end while

return \mathcal{S}_{next}

する。

- Step 5: Step 4 で得られた日本語列から語尾活用された語を基本形に変換する。

Step 1 では語の活用形を基本形にした日本語列への助詞の挿入に対して、Step 4 では語尾活用後の日本語列への助詞の挿入の違いがある。表 3 に表 1 で示した例のパイプラインによる翻訳例を示す。

表 3 パイプラインによる翻訳例

参照訳	年金について相談したいです。
\mathcal{S}_{-PP-C}	年金 について 相談 する たい です。
\mathcal{S}_{+PP-C}	年金 について 相談 する たい です。
\mathcal{S}_{+PP+C}	年金 について 相談 したい です。

4 モデルの学習

翻訳アルゴリズムで用いたモデルの学習方法について説明する。

4.1 統計的機械翻訳モデル

翻訳アルゴリズムの Step 0 には統計的機械翻訳 (SMT)[13] を用いた。学習は、対訳コーパスを用いて行ったが、目的言語である日本語は MeCab¹⁾ による形態素解析の結果から助詞を削除し、語尾活用のある語については、基本形に変換する前処理を行った。目的言語の言語モデルとしては、3-gram の Kneser-Ney smoothing を用いた。なお、学習データに gloss 辞書の定義を追加の対訳データとして含めた。これはコーパスサイズが少ないため学習データに一度しか現れない gloss が多く存在するためである。

1) <https://pypi.org/project/mecab-python3/>

4.2 seq2seq 機械翻訳モデル

翻訳アルゴリズムの Step 1 における機械翻訳モデル (m1) には, Transformer[14] による seq2seq 翻訳モデル [15] を用いた. 対訳コーパスは日本語コーパスに含まれる文から, 原言語, 目的言語を以下のようにして作成した.

- 原言語:原文の形態素解析の結果から助詞を削除し, 語尾活用のある語については, 基本形に変換
- 目的言語:原文の形態素解析の結果から語尾活用のある語については, 基本形に変換

これは単言語コーパスから \mathcal{S}_{-PP-C} から \mathcal{S}_{+PP-C} へ翻訳する擬似的な対訳コーパスを生成することを意味する. seq2seq のハイパーパラメータについては, 事前実験より以下のように定めた: encoder, decoder の embed サイズ:512, FFN のユニット数:2048, layer 数:6, attention head 数:8. また encoder の入力, decoder の出力には tied-embedding を使用した. Optimizer には Adam を使用し, 学習率は初期値 $5e-4$ で更新回数 の逆平方根に基づき減衰させた. 作成した対訳コーパスを 8:1:1 の割合で, train,dev,test データセットに分割し, dev による性能が一番高いパラメータで評価を行った.

翻訳アルゴリズムの Step 2 における機械翻訳モデル (m2), Step 4 における機械翻訳モデル (m3) は同様に Transformer ベースの seq2seq を用いたが, 擬似対訳コーパスの生成方法のみが異なる. 予備実験の結果から得られたパラメータと学習方法は機械翻訳モデル (m1) と同様であるため詳細は割愛する.

5 実験と結果

対訳コーパスからランダムに 10 個の対訳データをテストに選び, 残りを学習データとして, 統計的機械翻訳モデルを再学習したモデルで翻訳を行う実験を 100 回繰り返し, その結果得られた 1000 個の翻訳と, 参照訳とで翻訳性能の評価を行った. なお, seq2seq 翻訳モデル (m1,m2,m3) については, すべての実験で事前に学習済みのパラメータを使用し, ビームサイズ 5 でデコードを行なった.

比較するベースラインモデルとして以下の 4 つのモデルと, Yin et al. [12] が提案している end-to-end 手話機械翻訳モデル **G2T** (Gloss to Text) との比較実験を行った. Algorithm 1 で示した提案手法は

SMT+Iterative_seq2seq とした.

- **naive**: gloss 辞書を用いて, gloss を日本語へ置き換えることで翻訳を行うモデル. 辞書に複数の日本語が定義されている場合, 一番最初の語を選択する.
- **SMT**: 統計的機械翻訳モデルで \mathcal{S} から直接 \mathcal{S}_{+PP+C} へ翻訳を行うモデル.
- **SMT+1step_seq2seq**: Step 1, Step 2 の seq2seq 翻訳モデル (m1,m2) で行っている \mathcal{S}_{-PP-C} から \mathcal{S}_{+PP+C} への翻訳を一つの seq2seq で翻訳を行い, 逐次的に翻訳結果の更新を行わないモデル.
- **SMT+2step_seq2seq**: Step 0 から Step 2 までの処理を行い, 逐次的に翻訳結果の更新を行わないモデル.

G2T[12] はドイツ手話またはアメリカ手話のサイズの大きい対訳コーパス [10],[6] を使用している為, 予備実験において日本語対应手話-日本語の対訳コーパスを用いて, ハイパーパラメータ検索を行い dev セットにおいて一番性能が良かった以下のパラメータを用いて実験を行った. embed サイズ:256, FFN の embed サイズ:1024, layer 数:1, attention head 数:4 (encoder, decoder 共通)

性能評価は, BLEU1/2/3/4↑[16], METEOR↑[17], WER(Word Error Rate)↓を用いた. 結果を表 4 に示す. 結果より, gloss 辞書により単純に gloss を日本語へ置き換える翻訳手法では, 文法的に正しい日本語にならず低い性能を示す. G2T の性能はいずれの指標においても SMT の結果よりも低い値を示した. 手話言語の違いもあるが, 低リソースの場合, seq2seq モデルでは SMT の性能を超えることが困難であることを示している. SMT と seq2seq によるパイプラインモデルでは, SMT の性能から大幅に性能を向上している. これは日本語コーパスから作成した擬似対訳コーパスで学習された seq2seq モデルが, 手話翻訳における助詞の挿入, 語尾活用に非常に有効であることを示している. 更に表 5 に SMT とパイプライン翻訳モデルの BLEU3, BLEU4 スコア値による t-test 検定の結果を示した.

表 5 からパイプラインモデルが SMT よりも有意に性能が高いことがわかる. SMT+1step_seq2seq と SMT+2step_seq2seq とでは有意な差はなかったが, 提案手法である SMT+Iterative_seq2seq は BLEU3 において SMT+1step_seq2seq, SMT+2step_seq2seq 共

表 4 各モデルの翻訳性能の結果 (BLEU1/2/3/4↑, METEOR↑, WER↓)

	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	WER
naive	0.184	0.049	0.007	0.002	0.139	0.801
G2T (Yin et al. [12])	0.695	0.640	0.592	0.535	0.708	0.305
SMT	0.788	0.724	0.663	0.599	0.800	0.233
SMT+1step_seq2seq	0.810	0.752	0.697	0.638	0.830	0.301
SMT+2step_seq2seq	0.811	0.756	0.701	0.642	0.829	0.245
SMT+Iterative_seq2seq	0.817	0.762	0.707	0.648	0.833	0.216

表 5 SMT, SMT+1step_seq2seq, SMT+2step_seq2seq, SMT+Iterative_seq2seq の BLEU3, BLEU4 スコア値による t-test 検定の結果. 数値は p-value を示す. *は p-value が<0.05, **は p-value が<0.01 を示す.

	BLEU3			BLEU4		
	SMT	SMT+ 1step_seq2seq	SMT+ 2step_seq2seq	SMT	SMT+ 1step_seq2seq	SMT+ 2step_seq2seq
SMT+1step_seq2seq	<10 ^{-4**}			<10 ^{-4**}		
SMT+2step_seq2seq	<10 ^{-4**}	0.329		<10 ^{-4**}	0.354	
SMT+Iterative_seq2seq	<10 ^{-4**}	0.007**	0.026*	<10 ^{-4**}	0.036*	0.101

表 6 SMT+Iterative_seq2seq により翻訳結果を得るまでに
行ったループ処理の回数とその頻度

ループ回数	頻度
1	723 (72.3%)
2	258 (25.8%)
3	4 (0.4%)
4	6 (0.6%)
6	1 (0.1%)
7	2 (0.2%)
10>	6 (0.6%)

に, また BLEU4 については SMT+1step_seq2seq と有意な差を認めた. このことから, 助詞の挿入と語尾活用の推測には相互依存性があり逐次的に双方を修正する SMT+Iterative_seq2seq は翻訳精度を高める上で有効な手法であると判断できる. 次に, SMT+Iterative_seq2seq による翻訳が Algorithm 1 の何回目のループで収束したかを調査した. 結果を表 6 に示す. 表 6 より約 72%の結果については, ループ処理の回数が 1 であったことを示す. これは Algorithm 1 の Step 1 の語尾活用前の日本語列に対する助詞の推定と最初のループ処理での Step 4 における語尾活用後の助詞の推定が同じであったことを示している. また約 26%の結果については 2 回目のループ処理で結果が収束しており, これは語尾活用前の助詞の推定が最初の Step 4 における語尾活用後の助詞の推定で修正されたことを示している. ループ回数が 4 以上のものは, hallucination[18] エラーがあった場合で, その場合, 出力語列を切り詰めて出力するようにした.

付録の表 7 に各モデルが出力する翻訳例を表 8 に翻訳元となる gloss の辞書定義を示した. SMT, G2T

の翻訳には文法的な誤りを多く含むことが確認できる. $\delta_{+PP+C}(1)$, $\delta_{+PP+C}(2)$ は SMT+Iterative_seq2seq のループ処理における 1 回目, 2 回目の翻訳結果である. $\delta_{+PP+C}(1)$ で見られる助詞挿入の誤りが $\delta_{+PP+C}(2)$ では修正されていることがわかる. なお, 表 7 の最後の例のように助詞推定の曖昧性(「で」または「の」)が発生するケースもある. この対策として, 非手指動作 (NMS) を gloss の付属情報として用いることや, 会話の文脈情報をモデルに組み込んで判断することが考えられるが, 今後の課題としたい.

6 まとめ

本研究では, SMT により少量の対訳コーパスを用いて gloss を対応する日本語へマッピングする翻訳モデルと大量の日本語コーパスから得られた擬似対訳データにより学習した助詞の挿入を行う seq2seq 翻訳モデルと語尾活用を行う seq2seq 翻訳モデルをパイプラインで繋いだ手話から日本語への機械翻訳手法について提案した. 実験結果より, SMT のみを用いた場合に比べ BLEU3/4 による翻訳性能がそれぞれ+4.3/+4.9 向上することを確認した.

謝辞

本研究は, 電気通信大学とソフトバンク株式会社との「聴覚障害者と健聴者を繋ぐ相互コミュニケーションシステムに関する研究開発」の支援により実施された.

参考文献

[1]岡典栄, 赤堀仁美. 日本手話のしくみ. 大修館書店, 2011.

- [2]松岡和美. 日本手話で学ぶ手話言語学の基礎. くろしお出版, 2015.
- [3]Kazuyuki Kanda, Akira Ichikawa, Yuji Nagashima, Yushi Kato, Mina Terauchi, Daisuke Hara, and Masanobu Sato. Notation system and statistical analysis of NMS in JSL. In *Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, p. 181–192. Springer-Verlag, 2001.
- [4]Atsuko Koizumi, Hirohiko Sagawa, and Masaru Takeuchi. An Annotated Japanese Sign Language Corpus. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002.
- [5]Mayumi Bono, Kouhei Kikuchi, Paul Cibulka, and Yutaka Osugi. A Colloquial Corpus of Japanese Sign Language: Linguistic Resources for Observing Sign Language Conversations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pp. 1898–1904. European Language Resources Association, 2014.
- [6]Clayton Valli, Ceil Lucas, Kristin J. Mulrooney, and Miako N.P. Rankin. *Linguistics of American Sign Language*. Gallaudet University Press, Washington, DC, 2011.
- [7]松本忠博, 池田尚志. 日本語テキストから手話テキストへの機械翻訳の試み. 自然言語処理 / 言語処理学会 編, Vol. 15, No. 1, pp. 23–51, 2008.
- [8]Taro Miyazaki, Yusuke Morita, and Masanori Sano. Machine Translation from Spoken Language to Sign Language using Pre-trained Language Model as Encoder. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pp. 139–144. European Language Resources Association, 2020.
- [9]Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Sign language recognition with transformer networks. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6018–6024. European Language Resources Association, May 2020.
- [10]N. C. Camgöz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural Sign Language Translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7793, 2018.
- [11]N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10020–10030, 2020.
- [12]Kayo Yin and Jesse Read. Better Sign Language Translation with STMC-Transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5975–5989. International Committee on Computational Linguistics, December 2020.
- [13]Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- [14]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. 2017.
- [15]Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [16]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*, pp. 311–318. Association for Computational Linguistics, July 2002.
- [17]Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72. Association for Computational Linguistics, June 2005.
- [18]Chaojun Wang and Rico Sennrich. On Exposure Bias, Hallucination and Domain Shift in Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3544–3552, Online, July 2020. Association for Computational Linguistics.

付録

表7 gloss 辞書により \mathcal{G} を日本語単語へ置き換える naive な手法, G2T[12], 統計的機械翻訳 (SMT) と提案手法 SMT+Iterative_seq2seq による翻訳の例. \mathcal{S}_{-PP-C} は SMT による Step 0 の出力を示し, $\mathcal{S}_{+PP+C}(1)$, $\mathcal{S}_{+PP+C}(2)$ はそれぞれ seq2seq モデル (m1,m2,m3) により翻訳結果を 1,2 回更新した後の翻訳結果. いずれも更新回数 2 回で翻訳結果が収束している例.

		BLEU1	BLEU2	BLEU3	BLEU4	METEOR	WER
\mathcal{G}	g467 g12 g77						
参照訳	私の子供を探します。						
naive	私子供探す	0.176	0.0	0.0	0.0	0.152	0.714
G2T	私を探します。	0.67	0.581	0.532	0.474	0.712	0.286
SMT	私子供を探しています。	0.75	0.567	0.377	0.0	0.792	0.429
\mathcal{S}_{-PP-C}	私子供探すます。	0.536	0.3	0.0	0.0	0.464	0.429
$\mathcal{S}_{+PP+C}(1)$	私も子供を探します。	0.857	0.756	0.7	0.643	0.841	0.143
$\mathcal{S}_{+PP+C}(2)$	私の子供を探します。	1.0	1.0	1.0	1.0	0.999	0.0
\mathcal{G}	g19 g20 g87 g294 g20 g307 g202 g9						
参照訳	医療費の自己負担額は安くなりますか？						
naive	医療 お金 自分 負担 お金 安い なる か	0.227	0.0	0.0	0.0	0.129	0.75
G2T	医療 費 の 自己 負担 割合 は なる ます か ？	0.83	0.728	0.646	0.551	0.829	0.167
SMT	医療 費 の 自己 負担 の 安い なる ます か ？	0.75	0.691	0.62	0.531	0.717	0.25
\mathcal{S}_{-PP-C}	医療 費 自己 負担 費 安い なる ます か ？	0.573	0.457	0.277	0.0	0.538	0.417
$\mathcal{S}_{+PP+C}(1)$	医療 費 は 自己 負担 費 が 安くなりますか？	0.833	0.674	0.515	0.417	0.781	0.25
$\mathcal{S}_{+PP+C}(2)$	医療 費 の 自己 負担 費 は 安くなりますか？	0.917	0.866	0.807	0.735	0.895	0.08
\mathcal{G}	g73 g475 g52 g19 g27 g151 g2						
参照訳	障害者向け医療支援を知りたい。						
naive	障害者 会う 医療 助成 わかる 欲しい	0.322	0.201	0.0	0.0	0.29	0.667
G2T	障害者 向け 医療 支援 を 知りたい。	1.0	1.0	1.0	1.0	0.999	0.0
SMT	障害者 向け 医療 支援 を 知りたい。	1.0	1.0	1.0	1.0	0.999	0.0
\mathcal{S}_{-PP-C}	障害者 向け 医療 支援 知る たい。	0.772	0.698	0.599	0.525	0.777	0.222
$\mathcal{S}_{+PP+C}(1)$	障害者 向け の 医療 支援 を 知りたい。	0.9	0.837	0.759	0.658	0.984	0.111
$\mathcal{S}_{+PP+C}(2)$	障害者 向け 医療 支援 を 知りたい。	1.0	1.0	1.0	1.0	0.999	0.0
\mathcal{G}	g258 g860 g24 g8 g33 g9						
参照訳	老人ホームの情報をいただけますか？						
naive	老人 住宅 情報 いただく できる か	0.303	0.0	0.0	0.0	0.172	0.667
G2T	老人 ホーム の 情報 は いただけますか？	0.889	0.816	0.725	0.597	0.882	0.111
SMT	老人 住宅 へ の 情報 を いただけますか？	0.8	0.73	0.693	0.661	0.872	0.222
\mathcal{S}_{-PP-C}	老人 ホーム 情報 いただける ますか？	0.644	0.492	0.331	0.0	0.639	0.333
$\mathcal{S}_{+PP+C}(1)$	老人 ホーム で 情報 を いただけますか？	0.889	0.816	0.725	0.661	0.882	0.111
$\mathcal{S}_{+PP+C}(2)$	老人 ホーム の 情報 を いただけますか？	1.0	1.0	1.0	1.0	0.999	0.0

表8 表7で使われている gloss の辞書定義

g12	子供、児童、子	g258	老人、おじいさん	g467	私	g860	住宅、宿
g19	医療	g27	助成、支援、補助	g475	者、民	g87	自分、自己、自
g2	欲しい、好き、希望	g294	負担	g52	会う、向け、対する	g9	か、あなた
g20	お金、費用、運賃	g307	安い	g73	障害、壊す、破損		
g202	なる	g33	できる、大丈夫、よろしい	g77	探す、観光、見学		
g24	情報	g4	登録	g8	いただく、もらう		