日英ニューラル機械翻訳における未知語への対応

伊部 早紀　　松田 源立　　山口 和紀
東京大学
{ibe, matsuda, yamaguch}@graco.c.u-tokyo.ac.jp

1 はじめに

ニューラル機械翻訳 [2] は 2010 年代に提案された翻訳手法であり，原言語あるいは目的言語の単語を分散表現という数値ベクトルに変換し，ニューラルネットワークを用いてモデルを学習し翻訳を行う．従来の統計的機械翻訳 [6] では対訳コーパスに加えて単語アライメント情報などを用いて原言語から目的言語への変換規則の確率を学習していたが，この翻訳方式では対訳コーパスのみを入力として学習するため，以前よりもモデルが単純になった．

ニューラル機械翻訳は統計的機械翻訳に比べ流暢な文が生成されるが，訳抜けや重複など正確さに欠けることや，出力結果に未知語 (UNK) が含まれること [7] がしばしば指摘される．未知語の問題に対処する方法としては，コーパスに前処理を行う方法 [7, 10] や，学習するモデルを変更する方法 [1, 12] や，統計的機械翻訳と組み合わせる手法 [11] などがある．

本研究では，ニューラル機械翻訳で生成されるアテンションを用いて単語アライメントを作成し，それをもとに出力結果に含まれる未知語を他の単語に置き換えることで翻訳精度を向上させる手法を提案する．このように組み合わせることで，ニューラル機械翻訳による文の流暢さを残しつつ，単語アライメントを用いることで正確な単語を選択することが期待できる．

2 背景

2.1 フレーズに基づく統計的機械翻訳

本研究では，未知語を置き換える単語を探す時にフレーズベース機械翻訳 [6] で生成されるフレーズテーブルを使うため，フレーズベース機械翻訳について説明する．フレーズベース機械翻訳は，統計的手法を用いて翻訳規則を学習し，翻訳を行う手法である．手順を以下に示す．

第一に，対訳コーパスにアライメントモデルを適用し，単語アライメント表を作成する．アライメントモデルとしては IBM モデル 4 を使うのが一般的であり，本研究でも IBM モデル 4 を用いた．IBM モデル 4 の計算では同時に単語翻訳確率 $t(e|f)$ も計算される．単語アライメント表を原言語から目的言語，またその逆に対して作成し，grow-diag-final-and (gdfa) などのヒューリスティックを用いて 2 つの表を重ね合わせたものを最終的な単語アライメント表とする．

第二に，作成した単語アライメント表と対訳コーパスをもとに，翻訳規則とフレーズ翻訳確率 $P(\boldsymbol{e}|\boldsymbol{f})$ を学習する．

最後に，翻訳規則を利用して出力候補文をいくつか生成する．各候補文に対し翻訳規則の確率と言語モデルを用いて翻訳確率を計算し，翻訳確率が最も高い候補文を出力結果とする．

統計的機械翻訳では，単語やフレーズ間の対応関係は正確に抽出できるが，文全体の文法構造に関しては長文だと上手く翻訳できない．

2.2 アテンションに基づくニューラル機械翻訳

アテンションに基づくニューラル機械翻訳 [2] は，アテンションを用いて翻訳する原言語の部分を特定するニューラル機械翻訳である．

入力文 $\boldsymbol{f} = (f_1, f_2, ..., f_J)$ とその分散表現 $\boldsymbol{x} = (x_1, x_2, ..., x_J)$，出力文 $\boldsymbol{e} = (e_1, e_2, ..., e_I)$ とその分散表現 $\boldsymbol{y} = (y_1, y_2, ..., y_I)$ のペアの条件付き確率を最大化するように学習する．

$$p(\boldsymbol{e}|\boldsymbol{f}) = \prod_{t=1}^{I} p(e_t|e_1, ..., e_{t-1}, \boldsymbol{f}) \tag{1}$$

e_i の生成確率は以下で与えられる．

$$p(e_i|e_1, ..., e_{i-1}, \boldsymbol{x}) = \rho(y_{i-1}, s_i, c_i) \tag{2}$$

s_i は i 番目の入力単語列の隠れ状態であり，式 $s_i = f(s_{i-1}, y_{i-1}, c_i)$ で計算される．c_i は原文の i 番目の単語の文脈ベクトルといい，式 $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$ で計算する．α_{ij} はアテンション確率といい，x_i と y_i が関連している確率である．式 (3) に従って計算する．

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{Tx} \exp e_{ik}} \tag{3}$$

ここで $e_{ij} = v_a \mu(s_{i-1}, h_j)$ である．
h_j は入力文の j 番目の単語の文脈ベクトルで，式 $h_j = g(h_{j-1}, x_j)$ に従って計算する．
以上の $\mu(), \rho(), f(), g()$ はパラメータを含む非線形関数である．学習では，式 (4) のクロスエントロピーとして定義された損失関数を最小化し，翻訳では式 (1) を最大化する $\mathbf{e}$ をビームサーチなどで求める．

$$L = -\sum_j y_j \log y'_j \tag{4}$$

y_j はコーパスにおける y_j に対応する単語 e_j の出現確率、y'_j は学習したパラメータを用いて翻訳を行った時に y_j に対応する単語 e_j が出力される確率である．

ニューラル機械翻訳は長文でも文法構造が適切な文に翻訳できるが，未知語が多くなるという特徴がある．

3 提案手法

本研究ではニューラルネットワークを用いてモデルを学習し翻訳を行った後，アテンションから単語アライメントを作成し，作成した単語アライメントを用いて未知語を別の単語に置き換える手法を提案する．こうすることで，ニューラル機械翻訳の流暢さを残しつつ未知語を適切に減らすことができる．

3.1 単語アライメント表の作成

まず，出力文中の各未知語が原言語におけるどの単語に対応付けられているかを知るために，単語アライメント表を作成する．単語アライメントの作成アルゴリズムとしては intersection と gdfa の 2 種類を用いた．[6] を参考に，既存の手法である intersection と gdfa をアテンションに適用できるように修正して用いた．

入力 アテンション $A = a_{ij} \in [0, 1]$

出力 単語アライメント $B = b_{ij} \in \{0, 1\}$

各 a_{ij} と b_{ij} をセルと呼ぶ．またアテンションの値は確率と解釈する．

intersection 原言語，目的言語どちらから見ても確率が最も高いセルを対応付けする．

$$b_{ij} = \begin{cases} 1 & \text{if } i = \arg\max_{i'} a_{i'j} \text{ and } j = \arg\max_{j'} a_{ij'} \\ 0 & \text{otherwise} \end{cases}$$

gdfa intersection において作成した行列 B をもとに，対応付けされたセルの隣のセルで，原言語または目的言語のどちらか一方から見て確率が最も高いセルがあれば対応付けする．原言語の 1 つの単語が目的言語において複数の単語に対応している場合，それらは隣り合うことが多い．このため，すでにアライメント表に入っている単語と隣り合う単語のみを表に追加することを許す．

$$neighbor(b_{p,q}) = \{b_{p-1,q}, b_{p+1,q}, b_{p,q-1}, b_{p,q+1}\}$$

$$b_{ij} = \begin{cases} 1 & \text{if } b_{ij} = 0 \text{ and } \sum_{b_{pq} \in neighbor(b_{i,j})} b_{pq} \leq 1 \\ & \quad \text{and } (i = \arg\max_{i'} a_{i'j} \text{ or } j = \arg\max_{j'} a_{ij'}) \\ 0 & \text{otherwise} \end{cases}$$

3.2 未知語の置き換え

作成したアライメント行列 B をもとに，出力文における未知語 e_i が入力文のどの単語あるいは単語列 $\boldsymbol{f_i}$ に対応しているかを決める．
for all e_i in $\boldsymbol{e}$:

$$\boldsymbol{f_i} = \{f_j | b_{ij} = 1\}$$

次に，$\boldsymbol{f_i}$ の訳を決め，未知語 e_i と置き換える．

本研究では $\boldsymbol{f_i}$ の訳の決め方として [6] を参考に IBM と ChangePhrase の 2 種類を，さらに比較対象として Dict を加え計 3 種類の手法を用いた．

IBM 対訳コーパスに IBM モデル 4 を適用したときの単語翻訳確率 $t(e|f)$ を参照し，$\boldsymbol{f_i} = (f_{i_1}, ..., f_{i_n})$ の要素それぞれにおいて最も確率の高い $e_{best} = \arg\max_e t(e|f_i)$ を訳として選ぶ手法．

ChangePhrase $\boldsymbol{f_i}$ の訳を統計的機械翻訳で作成したフレーズテーブルから参照し，コーパスから計算したフレーズ翻訳確率 $P(\boldsymbol{e}|\boldsymbol{f}) = \frac{c(\boldsymbol{e},\boldsymbol{f})}{c(\boldsymbol{f})}$ が最も高い訳 $\boldsymbol{e_{best}} = \arg\max_{\boldsymbol{e}} P(\boldsymbol{e}|\boldsymbol{f_i})$ を未知語と置き換える手法．$c(\boldsymbol{f})$ はコーパス中のフレーズ $\boldsymbol{f}$ の出現回数，$c(\boldsymbol{e}, \boldsymbol{f})$ はフレーズ $\boldsymbol{e}$ と $\boldsymbol{f}$ の同時出現回数である．フレーズで置き換えるため，未知語を複数単語と置き換えることもある．

Dict 外部の辞書から f_j の各単語の訳を参照し，それを e_i と置き換える手法．

表 1: 各コーパスのサイズと単語数．train は学習に，dev はパラメータチューニングに，test はテストに用いた．

		train		dev	test
		文	単語	文	文
ASPEC	日本語	908.1K	162.3K	1.8K	1.8K
	英語		326.8K		
NTCIR	日本語	3.2M	146.5K	2.0K	0.9K
	英語		265.0K		

4 実験

4.1 実験データ・方法

対訳コーパスとしては ASPEC [8]，NTCIR10 の patentMT [5] を用いた．コーパスのサイズと単語数を表 1 に示す．モデル学習およびデコーダには nematus[1]を用い，日英および英日の翻訳を行った．英語の構文解析には Stanford Parser[2]を，日本語のトークン化には KyTea[3]を用いた．IBM モデル 4 の実装としては GIZA++[4]を用いた．フレーズテーブルの抽出は mosesdecoder[5]を用いた．未知語の置き換えの手法の Dict では外部辞書として EDict [4] を用いた．学習においては語彙数を頻度の高い 4 万語に制限した．

4.2 評価指標

翻訳精度の評価指標には BLEU [9]，METEOR [3] を用いた．BLEU はコーパス単位および文単位の双方で測り，文単位の評価値についてはベースラインと比較して平均値が上昇したものに関して片側検定における p 値を求めた．また METEOR は日本語の評価には対応していないため日英翻訳の評価のみに用いた．

4.3 実験結果と考察

nematus のデフォルトの設定で学習したモデルをベースラインとし，出力結果に提案手法による修正を加えたものと翻訳精度を比較した．実験結果を表 2 に示す．

表 2 より，日英翻訳においてはコーパス単位の BLEU は ASPEC では+intersection+IBM の場合が最大で+4.45，NTCIR では+gdfa+IBM の場合が最大で+1.18 の上昇，METEOR は ASPEC では+gdfa+ChangePhrase の場合が最大で+4.35，NTCIR では+intersection+IBM の場合が最大で+1.12 の上昇を確認できた．英日翻訳においてはコーパス単位の BLEU は ASPEC では +gdfa+ChangePhrase の場合が最大で+0.82，NTCIR では+gdfa+ChangePhrase の場合が最大で+0.77 の上昇を確認できた．英日翻訳より日英翻訳の方が未知語が多く出現するため，日英翻訳の方が BLEU 値が向上した．

また各コーパスともに UNK の出現回数が少なくなるとともに BLEU 値が向上することが確認できた．単語アライメント作成法は intersection，gdfa どちらも殆ど結果に差が見られないためにより単純な intersection を用いるのが良く，未知語置き換え法も ChangePhrase と IBM に差が見られないためにより単純な IBM を用いるのが良い．ChangePhrase は未知語に対応する単語が複数ある場合はそれらをフレーズとみなして訳を探すが，単語一つに対し一つの訳を決める IBM に比べてそれほど有効でないことがわかった．

5 おわりに

本研究では，生成されたアテンションをもとに，単語の重複がないように単語アライメントを作成することで，出力結果における未知語が入力文のどの単語に対応しているかを判別し，統計的機械翻訳でのモデルを用いて未知語を正しい単語に置き換えることで，翻訳精度の向上につながることを確認した．今後はモデルの変更を行い未知語の出現を減らす手法を検討していきたい．

参考文献

[1] P. Arthur, G. Neubig, and S. Nakamura. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, Vol. 29, pp. 65–72, 2005.

[1] https://github.com/EdinburghNLP/nematus
[2] https://stanfordnlp.github.io/CoreNLP/
[3] http://www.phontron.com/kytea/index-ja.html
[4] https://github.com/moses-smt/giza-pp
[5] https://github.com/moses-smt/mosesdecoder

表 2: 翻訳結果の精度評価．ベースラインの出力に修正を施した結果．太字は最も高い値．文単位の BLEU の平均値が baseline に比べ下がったものに関しては p 値を出していない．

ASPEC	日→英				英→日		
	BLEU		METEOR	UNK [%]	BLEU		UNK [%]
	コーパス単位	文単位			コーパス単位	文単位	
baseline	19.41	15.40	26.76	14.57	29.16	25.68	7.06
+intersection+ChangePhrase	23.06	19.06 ($p \leq e-9$)	30.46	6.15	29.49	25.94 (p=0.67)	2.48
+intersection+IBM	**23.86**	19.20 ($p \leq e-9$)	30.60	2.90	29.45	26.15 (p=0.45)	0.00
+intersection+Dict	19.91	15.99 (p=0.31)	27.86	10.93	29.22	25.66 (–)	1.73
+gdfa+ChangePhrase	23.41	**19.49** ($p \leq e-9$)	**31.11**	3.93	**29.98**	**26.33** (p=0.31)	1.06
+gdfa+IBM	23.24	19.25 ($p \leq e-9$)	30.97	2.87	29.54	26.07 (p=0.54)	0.00
+gdfa+Dict	19.97	16.06 (p=0.28)	28.02	9.95	29.24	25.61 (–)	2.49

NTCIR	日→英				英→日		
	BLEU		METEOR	UNK [%]	BLEU		UNK [%]
	コーパス単位	文単位			コーパス単位	文単位	
baseline	24.48	21.38	31.14	12.37	34.32	31.80	5.69
+intersection+ChangePhrase	25.03	21.87 (p=0.54)	31.55	7.88	34.76	32.18 (p=0.65)	3.28
+intersection+IBM	25.42	**22.79** (p=0.09)	**32.26**	0.00	35.01	32.34 (p=0.27)	0.00
+intersection+Dict	24.70	21.51 (p=0.87)	31.33	9.21	34.35	31.74 (–)	1.14
+gdfa+ChangePhrase	25.57	22.32 (p=0.24)	31.88	2.24	34.87	32.25 (p=0.59)	1.90
+gdfa+IBM	**25.66**	22.55 (p=0.15)	32.08	0.00	**35.09**	**32.62** (p=0.33)	0.00
+gdfa+Dict	24.81	21.54 (p=0.84)	31.44	5.54	34.37	31.72 (–)	1.69

[4] J. Breen. A www japanese dictionary. *Japanese Studies*, 20(3):313–317, 2000.

[5] I. Goto, K.-P. Chow, B. Lu, E. Sumita, and B. K. Tsou. Overview of the patent machine translation task at the ntcir-10 workshop. In *NTCIR*, 2013.

[6] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *NAACL HLT-Volume 1*, pp. 48–54. ACL, 2003.

[7] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.

[8] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In *Proc. of LREC 2016*, pp. 2204–2208.

[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pp. 311–318. ACL, 2002.

[10] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[11] I. Skadina and R. Rozis. Towards hybrid neural machine translation for english-latvian. In *Human Language Technologies–The Baltic Perspective: Proceedings of the Seventh International Conference Baltic HLT 2016*, Vol. 289, p. 84. IOS Press, 2016.

[12] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*, 2016.