大規模単言語資源を用いた大語彙ニューラル機械翻訳の検討

西村 友樹　　秋葉 友良　　塚田 元
豊橋技術科学大学
tnishi@nlp.cs.tut.ac.jp, akiba@cs.tut.ac.jp, tsukada@brain.tut.ac.jp

1 はじめに

機械翻訳は古くからよく研究されている分野の一つである。近年では、ニューラルネットワークを用いたニューラル機械翻訳 (NMT : Neural Machine Translation) が従来の統計的機械翻訳 (SMT : Statistical Machine Translation) よりも良い性能を示している。SMT はフレーズテーブルや言語モデルなど、個々に訓練された複数の要素から翻訳をモデル化するが、NMT は一つの大きなネットワークで翻訳プロセス全体をモデル化する。NMT はソース文からターゲット文への直接的なマッピングをモデルが学習することによって翻訳を行う。

一般的に NMT では単語を対応するビットが 1 でそれ以外が 0 の one-hot ベクトルで表現する。あらかじめ決めたボキャブラリに含まれない単語は特殊な UNK トークンとして表現する。通常、ボキャブラリは訓練セットに頻出するトップ K 単語で構築され、出現頻度の低いレアワードや訓練に出現しない単語は UNK トークンに置き換えられる。この手法はレアワードが訓練全体を妨害することを回避できるが、ボキャブラリサイズが小さいと多くの未知語を生成する問題がある。

本研究では大規模ボキャブラリを扱うために、大規模単言語資源で事前学習した単語の分散表現を用いる NMT のモデルを 4 つ提案する。実験では BLEU 尺度での改善は得られなかったが、分散表現を用いることでより適切な翻訳結果を出力できる事例を確認することができた。

2 関連研究

NMT の未知語に対処する研究の中で最もよく用いられる手法は、シーケンスの単位として単語よりも細かいサブワードを使用する手法である。バイトコード [1]、文字 [2]、BPE[3] などの様々なサブワードが検討されている。これらの手法は未知語をほぼゼロにすることができ、訓練データに出現しない単語であっても類似した構成要素を持つ単語が訓練に含まれていればその規則性を使って翻訳を実行できる。しかし、類似した構成要素を持つ単語が訓練データに含まれない単語に対応できないことや、単語に比べて入力シーケンスが長くなってしまうことでより長距離の依存を学習しなければならず訓練が困難になってしまう問題がある。

本研究では、未知語の問題を解決するために、入力や出力に大規模単言語資源で事前学習した分散表現を用いる NMT システムを提案する。提案法は単語区切りでシーケンスを分割するためシーケンスは長くならず、分散表現の持つ大規模ボキャブラリを直接利用する事でテストにのみ出現する単語にも対応できる。加えて、分散表現の類似単語が似たベクトルを持つ特徴により、出現頻度の低い単語であっても類似単語の学習に利用されることから、レアワード問題にも対処できる。

NMT の入力に分散表現を使用することは先行研究 (例えば [4][5]) でも行われており、新規のアイデアではないが、分散表現を使用することによる効果を調査した研究は少ない。また本研究では、事前学習した分散表現と NMT の学習によって得られる分散表現のそれぞれの利点を考慮した拡張手法を提案する。最近、低資源言語のための NMT として、単言語資源から学習した分散表現を NMT に利用する手法が Mattia ら [6] によって提案された。特に彼らの提案法のうち mix sum は、本研究の加算ハイブリッドモデルと同等である。彼らはヨーロッパ言語を対象に低リソース言語ペアについて分散表現の利用を検討したが、本研究では英日翻訳を対象に NMT の未知語問題への効果の観点から調査を行なった。

3 ニューラル機械翻訳

本研究ではエンコーダー、デコーダー、アテンションモジュールから構成されるアテンションベース NMT システム [7] を用いる。エンコーダーとデコーダーは Recurrent Neural Network で構成され、アテンションモジュールは feed-foward Neural Network で構成される。これら 3 つのモジュールはソース文が与えられた時にターゲット文の尤度が最大になるように協調しながら訓練される。

入力単語シーケンス $\mathbf{x} = x_1, x_2, \cdots, x_J$ とそれまでに出力した単語列 $y_1, y_2, \cdots, y_{i-1}$ が与えられるとモデ

ルは i 番目のターゲット単語 y_i を以下のように生成する。

$$P(y_i|y_1, y_2, \cdots, y_{i-1}, \mathbf{x}) = softmax(L_p(h_i)) \quad (1)$$

ここで $L_p(\cdot)$ は重みとバイアスを持つ非線形レイヤーで、h_i は i ステップでのデコーダー隠れ状態を表し、次のように計算する。

$$h_i = LSTM(h_{i-1}, d_{i-1}, a_i) \quad (2)$$

$LSTM(\cdot)$ は Long Short-Term Memory (LSTM) を表し、d_{i-1} は $i-1$ ステップで予測した単語のエンベディングを表す。エンベディングは簡単な線形変換で構築され、パラレルコーパスによって単語の表現を学習する。

$$d_{i-1} = W_d \cdot v_d(y_{i-1}) \quad (3)$$

ここで W_d は重みパラメータ、$v_d(y_{i-1})$ は $i-1$ ステップで出力した単語 y_{i-1} の One-hot ベクトルを生成する関数である。

a_i はアテンションモジュールによって計算されるアノテーションベクトルでソース表現の重み付け和によって計算される。アテンションモジュールでは Vaswani ら [8] が使用した scaled dot-product attention を採用し以下のように計算する。

$$\alpha_i = softmax\left(\frac{L_s(S) \cdot L_h(h_{i-1})^\top}{\sqrt{dim_a}}\right) \quad (4)$$

$$a_i = \alpha_i^\top \cdot S \quad (5)$$

ここで L_s, L_h は重みとバイアスを持つ線形レイヤーであり、dim_a は a の次元数を表す。S はエンコーダーの隠れ状態 $s_1, s_2, \cdots, s_J$ を表す行列でそれぞれの隠れ状態の計算は次のように計算する。

$$s_j = LSTM(s_{j-1}, e_j) \quad (6)$$

ここで e_j は入力単語列の j 番目の単語のエンベディングを表す。一般の NMT では、入力単語の one-hot ベクトル表現から、以下の式で線形変換をすることによりエンベディングを計算する。

$$e_j = W_e \cdot v_e(x_j) \quad (7)$$

一般の NMT では、式 7 のエンベディングを生成する行列 W_e もパラメータとして NMT の学習と同時に学習する。一方、one-hot ベクトルの次元数は学習前に決める必要があるため未知語の問題が生じる。本研究では、式 7 の代わりに、単言語コーパスから学習した分散表現から e_j を求める 4 つのモデルを提案する。

4 提案法

4.1 分散入力モデル

このモデルは分散表現を直接単語の特徴として利用するモデルである。分散表現は大規模単言語資源で事前学習されたベクトルであり、単語の特徴として利用できるだろうと考えた。具体的には式 6 の e_j について以下のように設定する。

$$e_j = Dist(x_j) \quad (8)$$

ここで $Dist(x_j)$ は単語 x_j の分散表現を取得する関数を表す。

4.2 分散線形モデル

分散表現は単言語のみで訓練されているため、パラレルコーパスで学習する翻訳タスクのための特徴として適していない可能性がある。そこで分散入力モデルに対して線形層を追加することでパラレルコーパスに適した表現を学習するモデルを提案する。式 8 を変形して

$$e_j = W_l \cdot Dist(x_j) + b_l \quad (9)$$

とする。ここで W_l, b_l は重み行列とバイアスパラメータを表す。

4.3 加算ハイブリッドモデル

one-hot ベクトルから構築したエンベディングは、NMT と同時に学習するために翻訳に適した分散表現が獲得されるが未知語に対応できない。一方、単言語コーパスで学習した分散表現は大語彙を扱えるものの翻訳に適切な表現ではない可能性がある。そこで両者を同時に入力するハイブリッド手法を提案する。加算ハイブリッドモデルでは 2 つの特徴量を加算することで単語 x_j の特徴量を計算する。式 6 を変形して次のように計算する。

$$f_j = W_l \cdot Dist(x_j) + b_l \quad (10)$$

$$g_j = W_e \cdot v_e(x_j) \quad (11)$$

$$e_j = \sigma(f_j + g_j) \quad (12)$$

$\sigma(\cdot)$ は非線形関数を表し、本研究では ReLU を採用した。

4.4 連結ハイブリッドモデル

連結ハイブリッドモデルでは連結した 2 つの表現から新しく特徴を生成する。このモデルでは 2 つの特徴量それぞれの有用な部分を選択してエンコーダー LSTM に特徴量として渡すことができるため性能の改

善が期待できる。特徴量は次のように計算する。

$$e_j = \sigma(W_c \cdot [f_j; g_j] + b_c) \tag{13}$$

5 実験

5.1 データセット

パラレルコーパスには 56,782 文からなる英日ロイターコーパスを使用し、テストセットと開発セットにそれぞれランダムに 2,000 文抽出し、残りを訓練セットに使用した。訓練は開発セットで最良のスコアを示したエポックで終了した。前処理として英語は Moses[9] のトークナイザーと小文字化処理を行い、日本語は MeCab を使った分かち書きと小文字化を行なった。

分散表現の訓練データとして英語 Wikipedia のダンプデータを取得し、wp2txt[10] を使って文章部分を抽出した。訓練は word2vec ツールを使い、skip-gram メソッドで学習した。窓幅 8、ネガティブサンプリング数 25、ベクトル次元数 500 で学習したところ、分散表現のボキャブラリサイズは 4M 単語であった。

NMT ベースラインとハイブリッドモデルのボキャブラリサイズは訓練セットに頻出するトップ 1K、3K、5K、10K 単語を選択し、それ以外の単語は 1 つの UNK トークンに置換した。また、比較のために単語区切りではなく BPE 区切りを採用したエンコーダーでの実験も行った。BPE のボキャブラリサイズは 10K に設定し、パラレルコーパスのソース文でバイトペアの訓練を行なった。エンコーダーは各モデル異なるネットワーク構造を持つが、デコーダーはボキャブラリサイズ 10K 単語を備えた共通のネットワーク構造を採用した。各モデルのボキャブラリサイズごとの未知語率を表 1 に示す。

表 1　比較した NMT モデルごとの未知語率 [%]

	ボキャブラリ	訓練	開発	テスト
source	top 1K	20.76	21.91	21.58
	top 3K	9.82	11.26	10.96
	top 5K	6.21	7.70	7.30
	top 10K	2.93	4.34	4.23
	BPE	0.00	0.01	0.01
	skip-gram	1.46	1.67	1.75
	hybrid (1K)	1.35	1.58	1.65
	hybrid (3K)	1.16	1.42	1.48
	hybrid (5K)	0.96	1.23	1.28
	hybrid (10K)	0.68	0.98	1.04
target	top 10k	0.88	1.39	1.20

表 2　ボキャブラリサイズごとの BLEU

	vocabulary size			
model	1k	3k	5k	10k
moses				19.92
baseline (単語)	17.80	**19.95**	**20.31**	**21.11**
baseline (BPE)	-	-	-	19.39
分散入力モデル				17.53
分散線形モデル				17.82
加算ハイブリッド	**19.33**	19.71	20.23	20.83
連結ハイブリッド	18.84	19.71	19.90	20.06

5.2 モデルパラメータ

エンコーダー RNN は 1 層の双方向 LSTM から構成され、デコーダー RNN は 1 層の単方向 LSTM で構成する。ワードエンベディングと LSTM の次元数は 500 次元とし、LSTM は正規分布からランダムに抽出した値で初期化した。ワードエンベディングレイヤーの重み W_e, W_d は事前学習した分散表現で初期化し、訓練によって更新した。

翻訳実験は英日翻訳タスクとし、Moses の multi-bleu.perl を用いて 4-gram BLEU を算出し比較した。また、SMT ベースラインとして 5-gram 言語モデルを含む Moses システムの結果も報告する。

5.3 翻訳性能評価

実験結果を表 2 に示す。最初に、ボキャブラリサイズの設定が必要なモデルについて 10K に設定した場合を比較するため、表 2 の最右の列に示されている結果に注目する。moses、分散入力モデル、分散線形モデルは、ボキャブラリサイズの設定が必要ないことに注意されたい。

まず、従来法と提案法の比較を行う。提案法（ハイブリッドモデル）は、単語をベースにした従来法（ベースライン (単語)）に性能は及ばなかったものの匹敵する性能を示した。これは BPE を用いた従来法や SMT よりも高く、未知語に対応したモデルの中では最も良い性能を示した。

次に提案法の間で性能比較を行う。翻訳のために単語の表現を更新する分散線形モデルがエンベディングを学習しない分散入力モデルよりも性能を改善していることがわかる。これは単言語で学習した表現では翻訳に適しておらず、パラレルコーパスで更新することでより高い性能を発揮できたことを示している。また、分散表現だけを使用するモデルよりもハイブリッドモデルの方が約 2 ポイント良い性能であることがわかる。2 つのボキャブラリを使うことで互いの未知語を補完して未知語を減らすことに成功したことや、似た単語の分散表現の入力に対しても one-hot の特徴と組み合わせることで性能が改善した。2 つのハイブリッドモ

表 3　翻訳例

ソース文	it said the move was to help control interest rates more **effectively** .
正解文	同中銀は、この措置は、金利をより効率的に管理するためとしている。
ベースライン (単語)	これによると、今回の措置は金利を管理させるためのものという。
分散線形	同中銀は、この措置は金利をより効率的に管理するためとしている。
ソース文	indiana’s soybean harvest pace advanced to 98 percent, up slightly from 92 percent the previous week, but behind last year’s and the five-year pace of 100 percent.
正解文	［ 大豆 ］＝収穫：今年 98 %、前週 92 %、前年同期 100 %、平年同期 100 %。
ベースライン (単語)	インディアナ州の大豆収穫は 90 %を上回っているが、前年同期に 90 %を下回った。
分散線形	[トウモロコシ] ＝収穫進度：今年 99 %、前週 53 %。
加算ハイブリッド	インディアナ州の大豆の収穫進度は、今年 19 %、前年同期比 98 %、平年同期より 60 %。

デルを比較すると、加算ハイブリッドモデルの方が性能が高い。

5.4　ボキャブラリ制限による影響調査

訓練セットから作成するボキャブラリのサイズを変更することで未知語率を変化させた実験を行なった (図 2 の 1k から 5k の列)。この制限の影響を受けるモデルはベースラインとハイブリッド手法である。分散入力モデルと分散線形モデルは、単言語資源から作成したボキャブラリを用いるためサイズは一定である。

ボキャブラリサイズを 1K に制限するケースでは、外部の大規模ボキャブラリを利用するハイブリッドアプローチが +1.53 ポイントと +1.04 ポイント、それぞれ性能を改善した。訓練セットから構築するボキャブラリサイズの制限が厳しい環境下において、外部ボキャブラリを利用して未知語に対処することが有効なアプローチであることを確認した。

5.5　翻訳例

実際のモデルで翻訳した例を表 3 に示す。ソース文の太字で表した単語はボキャブラリ制限によって単語ベースでは未知語として扱われることを示している。

最初の例では、ボキャブラリ制限によって”effectively”が未知語と扱われるため、従来法では”効率的”という部分が解釈できず意味が抜け落ちてしまっている。それに対して分散線形モデルは分散表現の大規模ボキャブラリによってモデルが”effectively”の特徴ベクトルを取り込むことができ、正しい翻訳を出力することができた。2 つ目の例はボキャブラリ制限によって未知語と扱われる単語はないが、分散線形モデルは”soybean”の翻訳として”トウモロコシ”を出力した。これはソース言語の分散表現空間において”soybean”と”corn”が類似したベクトルを持つことが原因である。加算ハイブリッドモデルはこのような単語においても one-hot 側の情報と組み合わせることで正しい訳”大豆”を翻訳することができた。また、分散表現を使った追加の情報を利用して、単語ベースでは出力できなかった”平年同期”なども出力することができた。

6　結論

ボキャブラリ制限を必要とする NMT に対して、大規模単言語資源を使って事前学習した分散表現によって大規模なボキャブラリを持つ NMT システムを提案した。英日翻訳実験では BLEU 尺度でベースライン性能を改善することはできなかったが、提案手法のハイブリッドモデルは、従来法を含む未知語に対応するモデルの中で最も高い性能を示した。また提案法により適切な翻訳を出力できる事例を確認できた。未知語を多く生成する設定では提案法がベースラインに勝る性能を達成した。

参考文献

[1] D.Gillick, C.Brunk, O.Vinyals, and A.Subramanya. Multilingual language processing from bytes. *CoRR*, Vol. abs/1512.00103, , 2015.

[2] A.Graves. Generating sequences with recurrent neural networks. *CoRR*, Vol. abs/1308.0850, , 2013.

[3] R.Sennrich, B.Haddow, and A.Birch. Neural machine translation of rare words with subword units. *CoRR*, Vol. abs/1508.07909, , 2015.

[4] T.Do, S.Sakti, and S.Nakamura. Toward expressive speech translation: A unified sequence-to-sequence lstms approach for translating words and emphasis, 08 2017.

[5] M.Artetxe, G.Labaka, E.Agirre, and K.Cho. Unsupervised neural machine translation. *CoRR*, Vol. abs/1710.11041, , 2017.

[6] M.Di Gangi and M.Federico. Monolingual embeddings for low resourced neural machine translation. 12 2017.

[7] D.Britz, A.Goldie, M.Luong, and Q. V.Le. Massive exploration of neural machine translation architectures. *CoRR*, Vol. abs/1703.03906, , 2017.

[8] A.Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A. N.Gomez, L.Kaiser, and I.Polosukhin. Attention is all you need. *CoRR*, Vol. abs/1706.03762, , 2017.

[9] H.Hoang, A.Birch, C.Callison-burch, R.Zens, R.Aachen, A.Constantin, M.Federico, N.Bertoldi, C.Dyer, B.Cowan, W.Shen, C.Moran, O.Bojar. Moses: Open source toolkit for statistical machine translation. pp. 177–180, 2007.

[10] Y.Hasebe. Method for using wikipedia as japanese corpus. *Doshisha studies in language and culture*, Vol. 9, No. 2, pp. 373–403, dec 2006.