階層的な部分単語を入力としたニューラル機械翻訳

Improving Neural Machine Translation by Incorporating Hierarchical Subword Features

森下 睦　　鈴木 潤　　永田 昌明
Makoto Morishita　Jun Suzuki　Masaaki Nagata

NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation

This paper focuses on the subword-based neural machine translation (NMT). We hypothesize that the appropriate subword-units for three modules in the NMT model, namely, (1) encoder embedding layer, (2) decoder embedding layer, and (3) decoder output layer, can differ each other. We empirically investigate our assumption, and find that incorporating several different subword-units for input and output embedding layers can consistently improve the BLEU score on the IWSLT 2012, 2013 and 2014 evaluation datasets.

1. はじめに

ニューラル機械翻訳とは，符号化器/復号化器モデル (Encoder-decoder model) または系列変換モデル (Sequence-to-sequence model)[Sutskever 14] と呼ばれる方法を用いた機械翻訳手法の総称として用いられている．ニューラル機械翻訳の研究は，世界中の著名な研究機関が多く参入し，人工知能/自然言語処理分野の国際会議や論文誌で数多くの研究成果が報告されている非常に活発な研究トピックである．

多くの研究報告の中から，いくつかの方法論はデファクトスタンダードの技術となり，後続の研究やツールで広く利用されている．最も広く取り入れられた技術の典型例としては，注意機構 (attention mechanism) があげられる [Bahdanau 15, Luong 15]．近年では，部分単語 (subword) 単位を用いる方法 [Sennrich 16, Schuster 12] がニューラル機械翻訳の研究では広く用いられるようになりつつある．本稿では，部分単語単位を用いる方法の中でも特に byte-pair encoding (BPE) を用いる方法 [Sennrich 16] に着目する．以降，特に注意書きがない限り，本稿での部分単語単位とは，BPE に基づいて決定された単位を意味することとする．

ニューラル機械翻訳において．部分単語単位を用いる利点はいくつか考えられるが，部分単語単位を用いる最も大きな理由は，言語生成時の未知語問題への対応である，従来のニューラル機械翻訳では，単語そのものを生成する方式で行なっていたため，基本的に学習データに出現しない単語を生成することは不可能であった．一方，部分単語単位を用いると，部分単語単位で構成できる単語は理論的には全て生成可能となり，システムが理論的に生成可能な語彙数を爆発的に増やすことが可能となる．一般的に，従来の単語単位のニューラル機械翻訳システムでは，生成できる語彙数は多くても数十万ぐらいが妥当な数で，数百万単位の語彙数でシステムを構築するには，かなり多くの計算機リソースと計算時間が求められることになる．この時，入出力データを部分単語単位に変換して用いるだけで，数千から数万の部分単語で，見かけ上数百万以上の単語を表現可能となり，未知語問題をおおよそ解決したと言える程の効果が得られる．また，副次的効果として計算リソースや計算時間の大幅削減にも貢献する非常によい性質を持っている．

連絡先: 森下 睦，NTT コミュニケーション科学基礎研究所，京都府相楽郡精華町光台 2-4, morishita.makoto@lab.ntt.co.jp

本稿では，このように優れた結果を実験的に示している部分単語単位に関して，更に一歩考察を進めていきたい．まず，前述のように，部分単語単位を導入した主な理由は生成時の未知語対応にある．つまり，復号化器の出力層の語彙を部分単語単位にすることが主な目的と言える．また，復号化器の出力層の語彙が決まると，当然入力層の語彙も同じものを使うことが自然であり一般的な方法論なので，こちらも同様に決定される．更に，復号化器だけを部分単語単位にすると不整合が生じる恐れがあるため一般的には，符号化器の語彙も復号化器の部分単語単位を構築する設定で構築する場合が多い．あるいは，通称 joint BPE [Sennrich 16] と呼ばれる入力文と出力文を一括して部分単語単位を求める方法が用いられることも多い．このように，ニューラル翻訳では，部分単語単位の関わる部位が符号化器の入力，復号化器の入力と出力と合計 3 種類存在するが，基本的に単一の部分単語単位を決定する方法を用いている．本稿では，この点に焦点をあて，符号化の入力単位，復号化器の入力単位と出力単位の計 3 種類の部分単語単位は，それぞれ適した部分単語単位は同じではないのではないかという仮説を立てる．つまり，これら 3 種類の部位は，当然違った役割でモデル内に組み込まれているのであり，その役割に則って効果的に機能する部分単語単位は違う可能性は否定できないが，これまでは特にこの点は指摘されず慣習に基づいて単一の設定で部分単語単位を決定していた．この仮説を検証するため，まず複数の部分単語単位が扱えるようにモデルを拡張する．その後，様々な数/長さの部分単語単位を作成し，部分単語単位を変えた場合の効果の違いを実験にて調査する．

2. 部分単語単位に基づくニューラル機械翻訳

本稿のベースラインモデルは，文献 [Luong 15] で用いられている注意機構（attention mechanism）付きの RNN 符号化/復号化器モデル [Bahdanau 15] とする．本モデルは，ニューラルネットに基づく翻訳や要約（ヘッドライン生成）タスクにおける強力なベースライン法という位置付けで，多くの論文で利用されている [Chopra 16, Kikuchi 16]．既に多くの参考文献が存在するので，本稿では紙面スペースの都合上，本稿の議論に直接関係ない部位に関しては詳細説明は割愛する．$\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^{I}$ を入力系列（入力文），$\boldsymbol{Y} = (\boldsymbol{y}_j)_{j=1}^{J}$ を出力系列

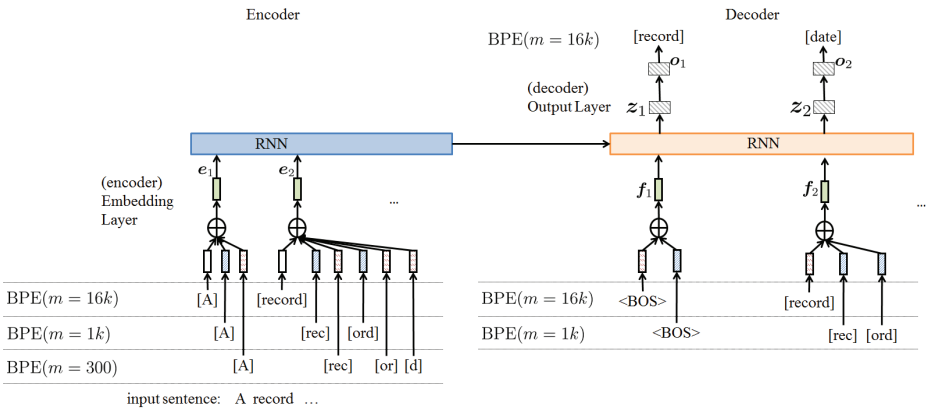

図 1: 提案法の簡単な構成図.

(出力文) とする [*1]. ただし, $\boldsymbol{x}_i$ を, i 番目の入力単語に対応する one-hot ベクトル表現, $\boldsymbol{y}_j$ を, j 番目の出力単語に対応する one-hot ベクトル表現とする [*2].

2.1 定式化

符号化器 (encoder): $\Omega^{(\mathrm{s})}(\cdot)$ を RNN で構成される符号化器の全ての処理を表す関数とする. 符号化器は, 入力 $\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^{I}$ を受け取って隠れ状態ベクトルのリスト $\boldsymbol{H}^{\mathrm{s}} = (\boldsymbol{h}_i^{\mathrm{s}})_{i=1}^{I}$ を返す処理となる.

$$\boldsymbol{e}_i = \boldsymbol{E}\boldsymbol{x}_i \tag{1}$$

$$\boldsymbol{H}^{(\mathrm{s})} = \Omega^{(\mathrm{s})}((\boldsymbol{e}_i)_{i=1}^{I}). \tag{2}$$

復号化器 (decoder) + 注意機構 (attention): 本稿では, K ベストビーム探索 (beam-search) を用いて, 入力系列 $\boldsymbol{X}$ が与えられたときの出現確率が最大となる出力系列 $\hat{\boldsymbol{Y}}$ (の近似解) を獲得する. ビーム探索では, 各処理時刻 j で K 個の出力候補を保持しながら探索を行う.

本稿では, 各処理時刻 j において, 生成する語彙を選択する処理を述べる. まず, 時刻 j における埋め込みベクトルの取得には, 以下の式を用いて計算を行う.

$$\boldsymbol{f}_{j,k} = \boldsymbol{F}\tilde{\boldsymbol{y}}_{j-1,k}, \tag{3}$$

この時, $\tilde{\boldsymbol{y}}_{j-1,k}$ は, 処理時刻 $j-1$ で予測された k 番目に確率が高い単語に対応する one-hot ベクトルとする. ただし, 全ての k に対して $\tilde{\boldsymbol{y}}_{0,k}^{(k)}$ は, 必ず特殊単語 BOS に対応する one-hot ベクトルとする.

次に得られた埋め込みベクトル $\boldsymbol{f}_j$ を使って, RNN と注意機構を用いて最終隠れ層のベクトル $\boldsymbol{z}_{j,k}$ を計算する.

$$\boldsymbol{z}_{j,k} = \texttt{RNN\_Attn}(u_{(j-1,k)}, \boldsymbol{f}_{j,k}, \boldsymbol{H}^{(\mathrm{s})}) \tag{4}$$

ここでは, `RNN_Attn` を入力ベクトル $\boldsymbol{f}_{j,k}$ を受け取って, RNN と注意機構を用いて最終隠れ層のベクトル $\boldsymbol{z}_{j,k}$ を計算する処理全てを表す関数とする. ただし, $u_{(j,k)}$ は, 処理時刻 j における k 番目の候補が, 処理時刻 $j-1$ の時の 1 から K 番目のどの候補から生成されたかを示す値とする. よって, $u_{(j,k)} = \{1, \ldots, K\}$ である. この値は, $j-1$ 時刻の時にどの RNN を用いて処理が行われたのかを次の時刻 j に伝達する役割を持っている. 次に, 得られた最終隠れ層のベクトル $\boldsymbol{z}_j$ から, 生成する単語を選択する基準となるスコアを以下の式を用いて計算する.

$$\boldsymbol{o}_{j,k} = \boldsymbol{W}^{(\mathrm{o})}\boldsymbol{z}_{j,k} + \boldsymbol{b}^{(\mathrm{o})}, \tag{5}$$

その後, K ベストビーム探索の処理を行い, 処理時刻 j における上位 K 個の候補を得る.

$$\{(\tilde{\boldsymbol{y}}_{j,k}, u_{(j,k)})\}_{k=1}^{K} = \texttt{BeamSearch}_k\Big((\boldsymbol{o}_{j,k})_{k=1}^{K}\Big), \tag{6}$$

ここで, K 個の候補とともに, 前述の $u_{(j,k)}$ の情報も取得する.

学習時は, $k = 1$ として予測結果 $\tilde{\boldsymbol{y}}_{j-1,k}$ の代わりに正解 $\boldsymbol{y}_{j-1,k}$ を利用することに相当する.

2.2 部分単語単位 (BPE)

統計量に基づく部分単語単位を決定する方法として, 既にいくつか提案がなされているが, byte-pair encoding (BPE) を用いた方法が昨今のニューラル機械翻訳の研究でよく見られる. BPE は入力文を最も細かい部品 (文字) まで分割し, 各部分単語 (文字を含む) を逐次的にマージ (結合) することで徐々に文字から単語へ部分単語を組み上げていく処理を行う. その組み上げの処理に置いて, 事前に決めたマージ回数に到達したら処理を終了する.

一般的には, マージ回数 (m) の値はハイパーパラメタであり, 人手により経験的に良いと思われる値が用いられる. 近年のニューラル機械翻訳では, 数千から数万の値が使われることが多く, 千以下, 十万以上の値はあまり用いられない傾向にある. これは, マージ回数が少ない場合は, 文字単位の処理に近く, それぞれの文字がもつ意味的な情報量は限定的になるため, あまり効果的ではないと予想され, また, マージ回数が多い場合は, 単語単位の処理と近くなり, 折角部分単語を導入した意味が薄れてしまうということが考えられる. このような理由から, 一般ドメインの翻訳で必要になる語彙数が数百万語彙だと仮定した場合, 経験的に数千から数万のマージ回数とするのは妥当な値と考えられる.

3. 提案法：階層的な部分単語の活用

BPE の特性として, マージ回数が 0 回の時は, 文字単位の処理と一致し, マージ回数を無限大にすると単語単位と同じになる. よって, BPE の観点で部分単語単位を用いる方法論を整理すると, 文字単位の処理から単語単位の処理までをマー

*1 簡単のため, 列ベクトルのリスト $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I)$ を, $(\boldsymbol{x}_i)_{i=1}^{I}$ と記述する.

*2 処理の単位は必ずしも単語である必要はないが, 説明を簡単にするために, ここでは処理の単位が単語であると仮定して議論をすすめる.

ジ回数という観点で，離散値で段階的に遷移する方法論と捉えることができる．つまり，BPE は，その性質上，文字単位の処理も単語単位の処理も包含する枠組みと捉えることができる．このことから，本稿では,「部分単語単位」という用語は，直感的に思い浮かぶ単語の一部という意味だけではなく，単語そのものや文字単位の状態も含む概念として用いる．また，本稿では，m を BPE のマージ回数を表す変数とし，特に，BPE($m = 0$) を文字単位を用いる方法，BPE($m = \infty$) を単語単位を用いる方法を表すこととし，以下の議論では，文字単位や単語単位の場合を区別せず全て部分単語の文脈で議論を行う．

図 1 に提案法の簡単な構成図を示す．提案法では，符号化器，及び，復号化器の入力層を拡張する．より具体的には，複数の部分単語単位を取り扱えるような拡張を行う．

まず，復号化器の出力に対して，複数の部分単語単位を出力するように修正することを考える．これはマルチタスク学習の設定と考えれば技術的には容易に対応可能であるが，符号化復号化器では，逐次的に単語予測を繰り返すという処理を行う性質上，複数の予測結果間の整合性を担保するには，制約付きの復号化処理などが必要となる．これは，学習と評価時の復号化（デコード）処理が煩雑になるため，本稿では取り扱わないこととする．よって本稿では，復号化器の出力部分は変更不要なことを担保した状態で符号化器，及び，復号化器の入力層の修正を行うという考えを基本方針とする．

この時，復号化器の入力部分の拡張は以下の通りである．

$$\boldsymbol{f}_{j,k} = \sum_r \boldsymbol{F}_r \psi_r(\tilde{\boldsymbol{y}}_{j-1,k}) \tag{7}$$

ただし，$\boldsymbol{F}_r$ は復号化器の埋め込み行列とする．前述のように，復号化器の予測は単一であることを仮定するため，$\psi_r(\tilde{\boldsymbol{y}}_{j-1,k})$ は，予測結果 $\tilde{\boldsymbol{y}}_{j-1,k}$ をキーとした事前に定義したマッピング関数を表しており，バイナリベクトルを返すこととする．また，$\psi_r(\cdot)$ によって返されるバイナリベクトルは，$\tilde{\boldsymbol{y}}_{j-1,k}$ の部分単語単位とする．例えば，BPE($m = 16k$) の `record` という部分単語が予測された場合，BPE($m = 1k$) で `record` の部分単語となる `rec` と `ord` がマッピング関数で引かれるといった処理となる．このように，より m が小さい部分単語単位は包含関係にあるため，一意に対象となる部分単語が決まり，容易に用いることができる．つまり，上記の復号化器の入力部分の拡張は，復号化器の予測結果が $\tilde{\boldsymbol{y}}_{j-1,k}$ 一つであるため，これからマッピング関数で一意に決定できる部分単語を，特徴として利用していることに相当する．

同様に，符号化器側の入力部分の修正は以下の通りである．

$$\boldsymbol{e}_i = \sum_q \boldsymbol{E}_q \phi_q(\boldsymbol{x}_i) \tag{8}$$

ただし，$\boldsymbol{E}_q$ は復符号化器の埋め込み行列とする．$\phi_q(\boldsymbol{x}_i)$ は，$\psi_r(\tilde{\boldsymbol{y}}_{j-1,k})$ と同様に，$\boldsymbol{x}_i$ から一意に導出可能なマッピング関数を表しており，要素が 0 または 1 をとるバイナリベクトルを返すこと仮定する．

4. 実験

ここでは，提案法の効果を検証するために，翻訳の実験を行う．

4.1 実験データ

本稿では，音声翻訳の評価型ワークショップである IWSLT[Cettolo 12] で利用されているデータを用いて評価実験を行った．また，複数ある言語対の中で，英語/フランス語，及び，英語/ドイツ語の両方向の翻訳，合計 4 種の実験設定を用いた．学習データには，IWSLT-2016 の学習データ (train)，開発データには，IWSLT-2012 の評価データ (tst2012)，評価データには，IWSLT-2013 と IWSLT-2014 の評価データ (tst2013, tst2014) をそれぞれ用いた．データの前処理として，Moses の tokenizer[*3] と truecaser[*4] の処理を行った．学習データに関しては，学習効率を考慮して 50 単語以上となる文を削除した [*5]．表 3 に用いたデータの規模を表す指標としてトークン数と文数を示す．これまでの議論の通り，部分単語単位への分割には，文献 [Sennrich 16] に対応する実装として公開されているコード [*6] を利用した．

実験に用いたモデルパラメタや学習の設定を表 4 に示す．学習時の学習率の調整方法として，最初の 30epoch は学習率の初期値 1.0 で学習を行い，その後 40epoch まで各 epoch 毎に学習率を 0.8 倍して減衰させて学習を行った．また評価時には，ビームサーチ（ビーム幅 20）を用いた．その際文長に対する正規化方法として，文献 [Cromieres 16] で紹介されているスコア（負の対数尤度）を単語数で割る方法を適用した．翻訳の評価としては，BLEU スコア [Papineni 02] を用いた [*7]．

4.2 結果

表 1 にドイツ語から英語 (DE-EN) と英語からドイツ語 (EN-DE) の翻訳実験結果を示す．同様に，表 2 にフランス語から英語 (FR-EN) と英語からフランス語 (EN-FR) の翻訳実験結果を示す．表中の `(a)` は近年最も標準的に用いられる部分単語単位を利用したニューラル機械翻訳の実験結果であり，本稿におけるベースラインと考える．`(b)` から `(h)` に関しては，`(a)` のベースラインから符号化器の設定を変更した実験結果である．同様に，`(i)` から `(k)` に関しては，`(a)` のベースラインから復号化器の設定を変更した実験結果である．最後に `(l)`, `(m)` は，符号化器と復号化器の両方の設定を変更した実験結果である.「単位」と「素性」の列が表しているものは，用いた部分単語単位のマージ回数 (m) である．複数記載がある場合は，複数の部分単語単位を一括して扱っていることを意味する．また ∞ は単語単位を意味する．カッコ内の値は，ベースライン `(a)` と比較した際の BLEU スコアの増減を表している．各評価データ（列）で最も高い BLEU スコアだったものを太字で示している．

実験結果から，通常の部分単語を入力とする手法と比較して，階層的な部分単語列を適用した提案法の方がより高い精度が得られていることがわかる．特に，符号化器に対して階層的な部分単語列を入力とした場合，ベースラインと比較して顕著に精度が向上している．さらに符号化器，復号化器ともに提案法を適用した場合でも全体的に精度が向上している．

これらの実験により，入出力層で適切な部分単語列は異なる可能性や，複数の部分単語列を階層的に入力とすることで既存

*3 `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl`

*4 `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl`

*5 ここでの 50 単語とは，Moses tokenizer の結果に対する単語数（トークン数）であり，BPE をかけた後の単語数ではない．BPE をかけたことによって単語数が 50 単語以上になっても特に削除などの処理は行わない．

*6 `https://github.com/rsennrich/subword-nmt`

*7 より詳細には，正解参照訳は前述の truecaser をかけた評価データを用い case-sensitive の設定で評価を行った．実際の評価スクリプトには，Moses に付属の `multi-bleu.perl` (`https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`) を用いた．

表 1: ドイツ語から英語 (DE-EN) と英語からドイツ語 (EN-DE) の翻訳実験結果

	Enc		Dec		DE-EN			EN-DE		
	単位	素性	単位	素性	tst2012	tst2013	tst2014	tst2012	tst2013	tst2014
(a)	16k	16k	16k	16k	31.70	34.38	28.81	26.00	28.67	24.05
(b)	∞	∞	16k	16k	29.99 (−1.71)	32.42 (−1.96)	27.39 (−1.42)	25.37 (−0.63)	27.65 (−1.02)	23.76 (−0.29)
(c)	∞	∞, 1k	16k	16k	32.20 (+0.50)	34.18 (−0.20)	29.72 (+0.91)	26.48 (+0.48)	29.09 (+0.42)	24.61 (+0.56)
(d)	∞	∞, 300	16k	16k	32.85 (+1.15)	34.84 (+0.46)	29.77 (+0.96)	26.93 (+0.93)	28.81 (+0.14)	25.18 (+1.13)
(e)	∞	∞, 1k, 300	16k	16k	32.60 (+0.90)	**35.22 (+0.84)**	30.08 (+1.27)	26.69 (+0.69)	29.03 (+0.36)	24.84 (+0.79)
(f)	16k	16k, 1k	16k	16k	31.78 (+0.08)	34.29 (−0.09)	29.17 (+0.36)	26.26 (+0.26)	28.85 (+0.18)	24.59 (+0.54)
(g)	16k	16k, 300	16k	16k	33.04 (+1.34)	34.90 (+0.52)	30.12 (+1.31)	26.72 (+0.72)	28.81 (+0.14)	24.29 (+0.24)
(h)	16k	16k, 1k, 300	16k	16k	**33.22 (+1.52)**	35.04 (+0.66)	**30.81 (+2.00)**	27.23 (+1.23)	29.47 (+0.80)	**25.35 (+1.30)**
(i)	16k	16k	16k	16k, 1k	31.59 (−0.11)	33.37 (−1.01)	28.87 (+0.06)	26.19 (+0.19)	28.81 (+0.14)	24.20 (+0.15)
(j)	16k	16k	16k	16k, 300	32.46 (+0.76)	34.43 (+0.05)	30.12 (+1.31)	26.13 (+0.13)	28.96 (+0.29)	24.21 (+0.16)
(k)	16k	16k	16k	16k, 1k, 300	31.47 (−0.23)	33.34 (−1.04)	29.10 (+0.29)	26.79 (+0.79)	28.78 (+0.11)	24.38 (+0.33)
(l)	∞	∞, 1k, 300	16k	16k, 1k, 300	32.69 (+0.99)	34.99 (+0.61)	29.47 (+0.66)	**27.41 (+1.41)**	**30.01 (+1.34)**	25.31 (+1.26)
(m)	16k	16k, 1k, 300	16k	16k, 1k, 300	32.86 (+1.16)	34.78 (+0.40)	30.38 (+1.57)	26.42 (+0.42)	29.19 (+0.52)	24.48 (+0.43)

表 2: フランス語から英語 (FR-EN) と英語からフランス語 (EN-FR) の翻訳実験結果

	Enc		Dec		FR-EN			EN-FR		
	単位	素性	単位	素性	tst2012	tst2013	tst2014	tst2012	tst2013	tst2014
(a)	16k	16k	16k	16k	42.24	39.56	36.67	43.99	40.00	36.83
(b)	∞	∞	16k	16k	40.54 (−1.70)	38.54 (−1.02)	36.22 (−0.45)	43.22 (−0.77)	39.43 (−0.57)	36.66 (−0.17)
(c)	∞	∞, 1k	16k	16k	42.63 (+0.39)	39.26 (−0.30)	36.93 (+0.26)	44.94 (+0.95)	40.68 (+0.68)	38.34 (+1.51)
(d)	∞	∞, 300	16k	16k	42.95 (+0.71)	39.79 (+0.23)	37.70 (+1.03)	**45.52 (+1.53)**	**41.79 (+1.79)**	38.23 (+1.40)
(e)	∞	∞, 1k, 300	16k	16k	43.11 (+0.87)	40.32 (+0.76)	**38.01 (+1.34)**	43.11 (−0.88)	40.32 (+0.32)	38.01 (+1.18)
(f)	16k	16k, 1k	16k	16k	43.27 (+1.03)	39.99 (+0.43)	37.28 (+0.61)	44.56 (+0.57)	40.94 (+0.94)	38.18 (+1.35)
(g)	16k	16k, 300	16k	16k	44.05 (+1.81)	40.20 (+0.64)	37.36 (+0.69)	44.94 (+0.95)	41.10 (+1.10)	38.50 (+1.67)
(h)	16k	16k, 1k, 300	16k	16k	**44.12 (+1.88)**	**40.40 (+0.84)**	37.42 (+0.75)	44.90 (+0.91)	40.68 (+0.68)	38.09 (+1.26)
(i)	16k	16k	16k	16k, 1k	42.37 (+0.13)	39.13 (−0.43)	37.23 (+0.56)	44.22 (+0.23)	40.21 (+0.21)	37.36 (+0.53)
(j)	16k	16k	16k	16k, 300	42.48 (+0.24)	38.78 (−0.78)	36.56 (−0.11)	43.45 (−0.54)	39.99 (−0.01)	37.13 (+0.30)
(k)	16k	16k	16k	16k, 1k, 300	42.50 (+0.26)	39.02 (−0.54)	26.62 (−0.05)	43.19 (−0.80)	40.29 (+0.29)	36.86 (+0.03)
(l)	∞	∞, 1k, 300	16k	16k, 1k, 300	42.98 (+0.74)	39.61 (+0.05)	37.43 (+0.76)	45.30 (+1.31)	40.88 (+0.88)	**38.54 (+1.71)**
(m)	16k	16k, 1k, 300	16k	16k, 1k, 300	43.71 (+1.47)	39.74 (+0.18)	37.53 (+0.86)	45.12 (+1.13)	41.32 (+1.32)	38.26 (+1.43)

表 3: 実験に用いたデータのトークン数と文数. (#token：トークン数. #sent：文数)

	DE-EN		FR-EN	
	#token	#sent	#token	#sent
train (学習データ)	3.2M	189.3K	3.8M	208.3K
tst2012 (開発データ)	30.9K	1.7K	21.7K	1.1K
tst2013 (評価データ)	21.0K	1.0K	21.9K	1.0K
tst2014 (評価データ)	25.0K	1.3K	25.0K	1.3K

表 4: 実験に用いたモデルと学習時のパラメタの値

ハイパーパラメタ	値
Embedding dimension	512
Hidden dimension	512
Attention dimension	512
Encoder layer	2
Decoder layer	2
Optimizer	SGD
Initial learning rate	1.0
Gradient clipping	5.0
Dropout rate	0.3
Mini-batch size	128

手法と比較してさらなる精度向上が見込めることが示せた.

5. おわりに

本稿では，近年デファクトスタンダードとなっている部分単語単位を用いたニューラル機械翻訳に着目し，符号化器の入力単位，復号化器の入力単位と出力単位の計 3 種類の部分単語単位は，それぞれ適した部分単語単位は同じではないのではないかという仮説を立て，それを実験的に検証することを主たる目的とした．この仮説を検証するために，本稿での提案法となる符号化器と復号化器の入力単位に複数の部分単語単位を同時に利用できるようなモデルの簡単な拡張を行った．IWSLT の実験データを用いた実験から，復号化器の出力単位から導出できる部分単語単位を利用することで，BLEU スコアによる翻訳性能が安定的に向上することが確認できた．同様に，符号化器は，元の単語の単位とマージ回数が少なめ ($m = 1k$ または $m = 300$) の部分単語単位を同時に利用することで，BLEU スコアによる翻訳性能が安定的に向上することが確認できた．また，これら二つを組み合わせることでも更に BLEU スコアによる翻訳性能が安定的に向上することが確認できた．

本稿での検証結果は，簡単な改良で安定的に翻訳性能を向上させることができることから，今後のデファクトスタンダードな方法論になりえることが期待できる．

参考文献

[Bahdanau 15] Bahdanau, D., Cho, K., and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, in *Proceedings of ICLR* (2015)

[Cettolo 12] Cettolo, M., Girardi, C., and Federico, M.: WIT3: Web inventory of transcribed and translated talks, in *Proceedings of EAMT*, pp. 261–268 (2012)

[Chopra 16] Chopra, S., Auli, M., and Rush, A. M.: Abstractive Sentence Summarization with Attentive Recurrent Neural Networks, in *Proceedings of NAACL*, pp. 93–98 (2016)

[Cromieres 16] Cromieres, F., Chu, C., Nakazawa, T., and Kurohashi, S.: Kyoto University Participation to WAT 2016, in *Proceedings of WAT*, pp. 166–174 (2016)

[Kikuchi 16] Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M.: Controlling Output Length in Neural Encoder-Decoders, in *Proceedings of EMNLP*, pp. 1328–1338 (2016)

[Luong 15] Luong, M.-T., Pham, H., and Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation, in *Proceedings of EMNLP* (2015)

[Papineni 02] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation, in *Proceedings of ACL* (2002)

[Schuster 12] Schuster, M. and Nakajima, K.: Japanese and korean voice search, in *Proceedings of ICASSP*, pp. 5149–5152 (2012)

[Sennrich 16] Sennrich, R., Haddow, B., and Birch, A.: Neural Machine Translation of Rare Words with Subword Units, in *Proceedings of ACL*, pp. 1715–1725 (2016)

[Sutskever 14] Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks, in *Proceedings of NIPS* (2014)