サブワード正則化: 複数のサブワード分割候補を用いたニューラル機械翻訳

工藤 拓　　Google 合同会社　taku@google.com

1　はじめに

ニューラルネットワークに基づく機械翻訳モデル (Neural Machine Translation, 以下 NMT) [4, 14, 12] では，目的言語生成時に語彙サイズに依存した計算量が必要であるため，事前に決めた比較的少量の語彙のみを使って学習することが多い．しかし，語彙の削減は，低頻度語を実質無視することを意味し，未知語の増加による翻訳精度の低下が避けられない．

バイトペア符号化 (Byte Pair Encoding, 以下 BPE) に代表されるサブワード化 [10, 14, 8] は，少量の語彙サイズのまま未知語の問題を解決する手法として多くの NMT システムで採用されている[1]．サブワードでは，高頻度語は 1 語として扱い、低頻度語はより短い単位 (部分文字列や文字) に分割される．どんな低頻度語も最終的には文字に分割されるため，未知語の問題が発生しにくい．さらに，BPE はコーパス中のサブワード数を最小化するように学習されるため，訳出時のステップ数がそれほど増加しない．サブワードは，語彙サイズとステップ数のバランスをうまくとる利点がある．

サブワード化により，テキストは一意のサブワード列に分割される．しかし，同一の語彙集合を用いた場合でも，サブワードの分割には表 1 のように複数の候補が存在する．NMT システムにとってはこれらは異なるサブワード列となるが，文生成タスクとして見た場合，これらの分割からは同一の文が生成される．つまり，これらは，文生成の精度に影響せず，曖昧性を解消する必要が本質的にない[2]．源言語の符号化においても，複数の分割を考慮することで，単語とその構成文字列の関係 (例えば，books は，book + s と構成される) を学習することが可能になり，分割の曖昧性やノイズに頑健になると考えられる．

本研究では，複数のサブワード分割を用いた新たな NMT の正則化手法 (**サブワード正則化**) を提案する．提案法は，2 つの手法から構成される．まず，複数のサブワード分割列を用いた NMT の学習手法を提案する．本手法は，学習データの動的なサンプリングとして定式化されるため，特定の NMT アーキテクチャに依存せず，既存の NMT のモデル構造に手を加える必要がない．次に，複数のサブワード分割候補を出力したり，それらからサンプリングできるよう，言語モデルに基づく新たなサブワード分割手法を提案する．

言語の異なる複数のコーパスを用いた翻訳実験により，単一の分割を用いる既存手法に比べ BLEU スコアが改善したことを示す．さらに，分野の異なる入力に対して，提案法が頑健に翻訳できることも確認した．

[1]2017 年の WAT 上位システムはほぼ BPE を採用している．

[2]BLEU による評価時には，出力文を評価用の単語分割器を使って再分割するため，BLEU の評価値も同一になる．

表 1: “Hello world” を表現可能な複数のサブワード列

サブワード列 (_は空白記号)	語彙 ID 列
_Hell o _world	13586 137 255
_H ello _world	320 7363 255
_He llo _world	579 10115 255
_ He l l o _world	7 18085 356 356 137 255
_H el l o _ world	320 585 356 137 7 12295

2　複数分割候補を用いた NMT

2.1　サンプリングによる学習

源言語文 X，目的言語文 Y に対し，$\mathbf{x} = (x_1, \ldots, x_M)$, $\mathbf{y} = (y_1, \ldots, y_N)$ をそれらのサブワード列とする．NMT は，翻訳確率 $P(Y|X) = P(\mathbf{y}|\mathbf{x})$ を式 1 でモデル化する．

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{n=1}^{N} P(y_n|\mathbf{x}, y_{<n};\theta) \tag{1}$$

ただし θ は，モデルのパラメータであり，$y_{<n}$ は，$n-1$ サブワードまで翻訳された部分翻訳列である．n 番目のサブワードの予測には，リカレントニューラルネットワーク (RNN) を用いることが一般的であるが，RNN を用いない手法も提案されている [12]．

学習データ $D = \{\langle X^{(s)}, Y^{(s)}\rangle\}_{s=1}^{|D|} = \{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)}\rangle\}_{s=1}^{|D|}$ に対し，パラメータ θ は，式 2 の最尤法で推定される．

$$\begin{aligned} \theta_{MLE} &= \arg\max_{\theta} \mathcal{L}(\theta) \\ \text{ただし } \mathcal{L}(\theta) &= \sum_{s=1}^{|D|} \log P(\mathbf{y}^{(s)}|\mathbf{x}^{(s)};\theta) \end{aligned} \tag{2}$$

文 X, Y が，確率 $P(\mathbf{x}|X), P(\mathbf{y}|Y)$ に従って複数のサブワード列に分割可能なとき，サブワード正則化では，式 3 の周辺尤度を用いて最適化を行う．

$$\mathcal{L}_{marginal}(\theta) = \sum_{s=1}^{|D|} \mathbb{E}_{\substack{\mathbf{x}\sim P(\mathbf{x}|X^{(s)}) \\ \mathbf{y}\sim P(\mathbf{y}|Y^{(s)})}} [\log P(\mathbf{y}|\mathbf{x};\theta)] \tag{3}$$

式 3 の学習は困難なため，k 個の分割を $P(\mathbf{x}|X), P(\mathbf{y}|Y)$ からサンプリングすることで尤度を近似する．

$$\begin{aligned} \mathcal{L}_{marginal}(\theta) &\cong \frac{1}{k^2} \sum_{s=1}^{|D|} \sum_{i=1}^{k} \sum_{j=1}^{k} \log P(\mathbf{y}_j|\mathbf{x}_i;\theta) \\ \mathbf{x}_i &\sim P(\mathbf{x}|X^{(s)}), \quad \mathbf{y}_j \sim P(\mathbf{y}|Y^{(s)}) \end{aligned} \tag{4}$$

本研究では，簡単のため，$k=1$ とする．ニューラルネットワークの学習には，ミニバッチ法を含むオンライン学習が用いられる．学習回数を十分増やすことで，サブ

ワード分割のサンプリングがオンライン学習の事例サンプリング経由で実行されるようになり，$k=1$ としても式 3 の良い近似を与える．ただし，パラメータの更新ごとに動的にサブワード分割をサンプリングすることに注意されたい．

2.2 デコード方法

デコード時には，源言語 X しか与えられない． 簡単には，$P(\mathbf{x}|X)$ が最大となる $\mathbf{x}^*$ を用いて翻訳を行えばよい．さらに，$P(\mathbf{x}|X)$ の n-best 解それぞれに対し，翻訳結果 $\mathbf{y}$ を求め，以下のスコアが最大となる翻訳を選択することも可能である．

$$score(\mathbf{x},\mathbf{y}) = \log P(\mathbf{y}|\mathbf{x})/|\mathbf{y}|^{\lambda} \tag{5}$$

ただし，$|\mathbf{y}|$ は $\mathbf{y}$ のサブワード数，$\lambda \in \mathbb{R}$ はサブワード数の影響を制御するパラメータであり開発データを用いて推定する． 本論文では，前者を one-best デコード，後者を n-best デコードと呼ぶ．

3 言語モデルによるサブワード分割

3.1 BPE の問題点

BPE[10] 及び wordpiece[8] は，1 文字 1 語彙から開始し，連結した際に最も頻度[3]が高くなる 2 つの語彙を選び新たな語彙とする手続きを決められた語彙サイズに達するまで繰り返すことで語彙結合ルールを学習する．分割は，語彙結合ルールを同一順序で適用することで行われる．BPE および wordpiece は，決定的アルゴリズムであるため，複数の分割を出力することは困難である．たとえ複数の分割が出力できたとしても，確信度の計算ができないため，解候補からサンプリングしたり n-best 解を出力する等の拡張が難しい．

3.2 ユニグラムモデル

本研究では，複数のサブワード分割を確信度付きで出力できる ユニグラムモデルに基づくサブワード分割方法を提案する．ユニグラムモデルでは，任意の分割 $\mathbf{x} = (x_1, x_2, \ldots, x_M)$ の分割確率 $P(\mathbf{x})$ を各サブワードの生起確率 $p(x_i)$ の積で表す[4](式 6)．

$$P(\mathbf{x}) = \prod_{i=1}^{M} p(x_i), \quad \forall i\ x_i \in \mathcal{V},\ \sum_{x\in\mathcal{V}} p(x) = 1 \tag{6}$$

ただし，$\mathcal{V}$ は，事前に与えられた語彙集合である．入力文 X に対する最適分割列 $\mathbf{x}^*$ は，全分割候補集合 $\mathbf{x} \in \mathcal{S}(X)$ から $P(\mathbf{x})$ が最大になる分割列をビタビアルゴリズムを用いて探索することで導出する (式 7)．

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{S}(X)} P(\mathbf{x}) \tag{7}$$

語彙集合 $\mathcal{V}$ が事前に与えられている場合，生起確率 $p(x_i)$ は，$p(x_i)$ を隠れ変数とする尤度 $\mathcal{L}$ の最大化によって求められる (式 8)．$\mathcal{L}$ による確率の推定には，EM アルゴリズムを用いる．

$$\mathcal{L} = \sum_{s=1}^{|D|} \log(P(X^{(s)})) = \sum_{s=1}^{|D|} \log\Big(\sum_{\mathbf{x}\in\mathcal{S}(X^{(s)})} P(\mathbf{x})\Big) \tag{8}$$

しかし，実際には $\mathcal{V}$ も未知である．そこで，十分大きなシード語彙集合から開始し，EM アルゴリズムによる確率推定と不要語彙の削除を交互に繰り返すことで学習を行う．具体的な手順を以下に示す．

1. 十分に大きなシード語彙 $\mathcal{V}$ を準備
2. $\mathcal{V}$ が決められたサイズになるまで以下を繰り返す
   (a) 生起確率 $p(x)$ を学習データと EM 法にて推定
   (b) 各語彙 $x \in \mathcal{V}$ に対し，x を削除したときの尤度の差分 (貢献度) を計算
   (c) 貢献度の小さい η% の語彙を削除[5] (ただし 1 文字が 1 語彙となる候補は残す)

シード語彙集合の作成にはいくつかの方法が考えられるが，本研究では，全文字集合と上位 100 万の高頻度部分文字列の和を用いる[6]．高頻度部分文字列は拡張接尾辞配列を用いて線形時間で列挙可能である [6].

BPE は，コーパス中のサブワード数 M の最小化を目的関数として貪欲に学習を行う辞書式データ圧縮手法である．一方，ユニグラムモデルは，対数尤度 $\sum \log p(x_i)$ の最大化を目的関数とするエントロピー圧縮手法の一種である．このことは，任意のシンボル s の出現確率を p_s とすると $-\log p_s$ の長さの符号語を割り当てた時に最短の符号となることから説明できる．テキスト圧縮という観点でこれら 2 つは共通しているが，ユニグラムモデルは言語モデルに基づいているため，尤度や確信度の計算，n-best 出力，サンプリング等の拡張が容易である．

3.3 ユニグラムモデルからのサンプリング

サブワード正則化では，分割候補から 1 つの分割を学習のイテレーションごとにサンプリングする．分割確率 $P(\mathbf{x}|X)$ からの近似的サンプリング手法として n-best 解の利用がある．具体的には，$P(\mathbf{x})\,(\mathbf{x} \in \mathcal{S}(X))$ の降順に l 個の分割結果 $(\mathbf{x}_1, \ldots, \mathbf{x}_l)$ を求め [5]，$P(\mathbf{x}_i|X) \cong P(\mathbf{x}_i)^{\alpha}/\sum_{i=1}^{l} P(\mathbf{x}_i)^{\alpha}$ をパラメータとする多項分布から $\mathbf{x}_i$ をサンプリングする．ただし，$\alpha \in \mathbb{R}^+$ は，確率分布のなめらかさを制御する逆温度パラメータである．

$l \to \infty$ とすれば全解空間からサンプリングが可能であるが，全解を陽に列挙することは計算量の観点から困難である．$l \to \infty$ のときには，Foward-filtering backward sampling 法 (以下 FFBS)[9] を用いる． FFBS では，まず，全解集合をラティスとして表し，文頭から各サブワードまでの前向き確率を求める．次に，文末から文頭方向にラティスをたどりながら，各分岐ごとに前向き確率にしたがってサブワードをサンプリングしていく．

[3] wordpiece の場合は尤度
[4] 目的言語 $\mathbf{y}$ も同様に表せる．紙面の都合上割愛する．
[5] 全実験で $\eta = 20$ とした．
[6] BPE を十分な結合回数適用することも考えられる．

4 関連研究

サブワード正則化は，学習データに対するノイズ付加と関連性がある．ノイズ付加はニューラルネットワークの正則化として広く知られおり，その代表例の dropout[11] は，ノードの一部を無作為に削除することで過学習を低減させる効果がある．サブワード正則化は，学習時にサブワード列を確率的に変化させるという意味において dropout と類似性がある．Denoising Autoencoder[13], (以下 DAE) は，入力層にノイズを加えることでノイズ除去するような学習がなされ汎化性能を向上させる技術である．自然言語処理においても，文の語順を無作為に変えた入力から元の文を復元するような DAE が教師なし NMT に応用されている [3, 1].

また，サブワード正則化は，データ拡張としてとらえることもできる．同一文を不変操作により複数のサブワード列に展開することは，画像認識における移動，回転，縮小・拡大等のデータ拡張と共通点がある．

5 実験

5.1 実験設定

複数の言語対，サイズの異なるコーパスにて本手法の有効性を BLEU[7] を用いて検証した．NMT システムとして GNMT[14] を用いた．GNMT は，Residual 結合付き複数層の LSTM 符号化，復号化による，注意機構付き NMT システムである．表 2 にコーパスの概要と使用したパラメータをを示す[7] [8] [9] [10] [11]．共通の設定として，dropout 確率は 0.2 とし，学習には Adam[2] と SGD を組み合わせた手法を用いた [14]．デコード時の文長正規化，被覆パラメータはともに 0.2 とした．IWSLT15, WMT14 コーパスでは，Moses トークナイザ[12]による前処理結果から，それ以外は生文から符号，復号化共通の語彙とサブワードモデルを構築した．ただし，空白をまたぐサブワードは語彙の対象外とした[13]．評価時の単語分割には，ja は KyTea[14], zh は文字による分割，それ以外は Moses トークナイザを用いた．

各コーパス，言語対において 7 つのシステムの比較を行った．ベースラインは，BPE 分割のみを用いる手法とした．提案手法については、サンプリング方法の異なる 3 種類の実験を行った．1 つは，ユニグラムモデルのベスト解のみを用いる手法 (l=1) であり，BPE とユニグラムモデルの比較を目的としている．ほかに予備実験結果から選択した (l=64, α=0.1)，(l=∞, α=0.5) のパラメータでサンプリングを行った (3.3 章参照)．また，デコード法として one-best と n-best デコードの比較を行った (2.2 章参照)．BPE は複数分割出力ができないため，one-best デコードの結果のみとなる．

さらに，提案手法の正則化の効果を検証するために，分野の異なる社内評価コーパス (ウェブ，特許，クエリログ) を用いた評価も行った．ただし，KFTT，ASPEC は，学習コーパスの分野が偏っておりベースラインの精度が低いため対象外とした[15].

5.2 実験結果と考察

表 3 に，各コーパスにおける BLEU スコアを示す．

まず，いずれのコーパス，言語対においても，BPE とユニグラムモデル (l = 1) は，ほぼ同程度の BLEU スコアが得られている．両サブワード手法ともテキスト圧縮を基礎とすることから，大きな精度差が出るとは考えにくく，妥当な結果だといえる．

サブワード正則化 (l > 1) により，WMT14(en→cs) 以外で BLEU スコアの向上が確認できる．とくにサイズの小さいコーパス (IWSLT15, KFTT) での効果は大きい．サブワード正則化によるデータ拡張が小規模コーパスでより大きく作用したものと考察できる．

サンプリング手法については，(l=64, α=0.1) の設定が全体的に高い精度を示しており，最適分割周辺のみから保守的にサンプリングするだけで正則化の効果は十分であることが確認できる．

また，n-best デコードにより，WMT14 コーパス以外でさらなる BLEU スコアの向上が確認できる．ただし，n-best デコードの使用にはサブワード正則化は必須であり，正則化なし (l = 1) の場合は，逆に精度が悪化するケースもある．正則化なしの場合，最適分割に特化した学習が行われ，複数の分割結果を効果的に扱えないことが精度低下の要因と考えられる．

表 4 に異なる分野のコーパスにおける結果を示す．標準データに比べ BLEU スコアの上昇が大きく，大規模コーパスの WMT14(en→cs) についても改善が確認できる．これらから，本手法が学習コーパスのサイズによらず分野の違いに頑健であることがわかる．

6 おわりに

本研究では，複数のサブワード分割を用いた NMT のための正則化手法 (**サブワード正則化**) を提案した[16]．複数の言語対，コーパスによる実験にて BLEU スコアの向上を確認した．本手法は，小規模のコーパスにおいてとくに有効性が高い．また，分野の異なるコーパスに対する頑健性も明らかになった．本手法は，言語非依存であり，既存 NMT モデルの変更が必要ないため適用範囲が広い．

今後の課題としては，まず，対話生成や自動要約等のニューラル言語生成への応用が挙げられる．これらの応用は，十分な学習データが確保できないことが多く，サブワード正則化によるデータ拡張が有効に機能する可能性が高い．さらに，データ拡張，ノイズ付加，テキストの自己符号器等の基礎技術基盤としてサブワード正則化の応用範囲を模索したい．

[7] IWSLT15: http://workshop2015.iwslt.org/
[8] KFTT: http://www.phontron.com/kftt/
[9] ASPEC: http://orchid.kuee.kyoto-u.ac.jp/ASPEC/
[10] WMT14: http://statmt.org/wmt14/
[11] WMT14(en↔de) は，文献 [14] と同一設定である．
[12] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
[13] 英語等では，頻度付き単語集合からのサブワード学習と同一になる．
[14] http://www.phontron.com/kytea/index-ja.html
[15] 京都に関するウィキペディア記事 (KFTT) と論文 (ASPEC) が対象であり，異分野の翻訳は困難であると判断した．
[16] https://github.com/google/sentencepiece から入手可能．

表 2: 評価用コーパスの概要

コーパス	言語対	文数 学習	開発	評価	パラメータ 語彙サイズ Enc/Dec 共有	LSTM, 埋め込み層次元数	LSTM レイヤ数 (Enc+Dec)
IWSLT15	en ↔ vi	133k	1553	1268	16k	512	2+2
	en ↔ zh	209k	887	1261	16k	512	2+2
KFTT	en ↔ ja	440k	1166	1160	8k	512	6+6
ASPEC	en ↔ ja	2M	1790	1812	16k	512	6+6
WMT14	en ↔ de	4.5M	3000	3003	32k	1024	8+8
	en ↔ cs	15M	3000	3003	32k	1024	8+8

表 3: 実験結果 (BLEU(%)) (l はサブワードのサンプリング対象数、α は逆温度パラメータ)

コーパス	言語対	ベースライン	提案法 (one-best デコード) $l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.5$	提案法 (n-best デコード, n=64) $l=1$	$l=64$ $\alpha=0.1$	$l=\infty$ $\alpha=0.5$
IWSLT15	en → vi	25.61	25.49	27.68	26.50	25.33	28.18	26.98
	vi → en	22.48	22.32	24.73	24.01	22.04	24.66	23.93
	en → zh	16.70	16.90	19.36	18.55	16.73	20.14	19.22
	zh → en	15.76	15.88	17.79	16.75	16.23	17.75	17.44
KFTT	en → ja	27.85	28.92	30.37	30.01	28.55	31.46	31.43
	ja → en	21.37	21.46	22.33	22.04	21.37	22.47	22.64
ASPEC	en → ja	40.62	40.66	41.24	41.23	40.86	41.55	41.87
	ja → en	26.51	26.76	27.08	27.14	27.49	27.75	27.89
WMT14	en → de	24.53	24.50	25.04	24.74	22.73	25.00	24.57
	de → en	28.01	28.65	28.83	29.39	28.24	29.13	29.97
	en → cs	25.25	25.54	25.41	25.26	24.88	25.49	25.38
	cs → en	28.78	28.84	29.64	29.41	25.77	29.23	29.15

表 4: 異分野コーパスによる評価結果 (BLEU(%)) ($l=64$, $\alpha=0.1$, one-best デコード)

分野	コーパス	言語対	ベースライン	提案法
ウェブ	IWSLT15	en → vi	13.86	16.51
		vi → en	7.83	10.08
		en → zh	9.71	12.73
		zh → en	5.93	8.71
	WMT14	en → de	22.71	26.02
		de → en	26.42	29.63
		en → cs	19.53	21.41
		cs → en	25.94	27.86
特許	WMT14	en → de	15.63	25.76
		de → en	22.74	32.66
		en → cs	16.70	19.38
		cs → en	23.20	25.30
クエリ	IWSLT15	en → zh	9.30	11.25
		zh → en	14.94	19.48
	WMT14	en → de	25.93	29.82
		de → en	26.24	30.90

参考文献

[1] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. *arXive preprint arXiv:1710.11041*, 2017.

[2] D. P. Kingma and J. B. Adam. A method for stochastic optimization. 2014. *arXiv preprint arXiv:1412.6980*.

[3] G. Lample, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. *arXive preprint arXiv:1711.00043*, 2017.

[4] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proc of EMNLP*, 2015.

[5] M. Nagata. A stochastic japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *In Proc. of COLING*, 1994.

[6] G. Nong, S. Zhang, and W. H. Chan. Linear suffix array construction by almost pure induced-sorting. In *In Proc. of DCC*, 2009.

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *In Proc. of ACL*, 2002.

[8] M. Schuster and K. Nakajima. Japanese and korean voice search. In *In Proc. of ICASSP*, 2012.

[9] S. L. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 2002.

[10] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proc. of ACL*, 2016.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1), 2014.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXive preprint arXiv:1706.03762*, 2017.

[13] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of ICML*, 2008.

[14] Y. Wu, M. Schuster, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.