後段モデルの損失値を用いた単語分割のタスクへの最適化

平岡達也 †, 高瀬翔 †, 内海慶 ‡, 欅惇志 ‡, 岡崎直観 †

† 東京工業大学 ‡ デンソーアイティーラボラトリ
{tatsuya.hiraoka@nlp., sho.takase@nlp., okazaki@}c.titech.ac.jp
{kuchiumi, akeyaki}@d-itlab.co.jp

1 はじめに

単語分割は自然言語処理の性能に影響を与える重要な処理である．単語分割にはルールベースの手法 [1] や辞書を用いた手法 [2]，教師なしの手法 [3, 4] が用いられており，単語分割の違いが後段タスクの性能の差に繋がることがわかっている [5, 6, 7, 8]．しかし，ある単語分割が後段タスクに適切かどうかは，実際に分割済みのコーパスで後段モデルを学習し性能を評価するまで分からないため，適切な単語分割を人手で選択することは難しい．

近年では，後段タスクに応じて単語分割を自動で最適化する手法が研究されている．Incremental-BPE [9] や DPE [10] は，機械翻訳タスクの性能向上に繋がるような単語分割を求める手法である．また，OpTok [11] は文書分類タスクにおいて，End-to-End に単語分割を最適化する手法である．しかし，これまでの研究は機械翻訳や文書分類のように想定タスクが限定されており，あらゆる後段タスクに適用可能な単語分割の最適化手法は存在していない．

そこで本稿では，ニューラルネットワークを用いたさまざまな後段タスクに適用可能な単語分割の最適化手法を提案する．提案手法は後段タスクの学習に用いるモデル (後段モデル) に単語分割器を組み合わせ，両者を同時に学習する．これにより，提案手法は学習コーパスや単語分散表現，モデルのパラメータなどのタスクに関わるあらゆる要素を考慮した単語分割の Emd-to-End な学習が可能である．

本稿では 3 言語での文書分類タスクにおける実験を通して，提案手法が既存手法である OpTok を上回る性能であることを確かめた．また，複数言語対での機械翻訳タスクにおける実験でも，提案手法は既存手法の DPE を上回る性能であり，提案手法が複数の後段タスクに適用可能な単語分割の最適化手法であることが確かめられた．

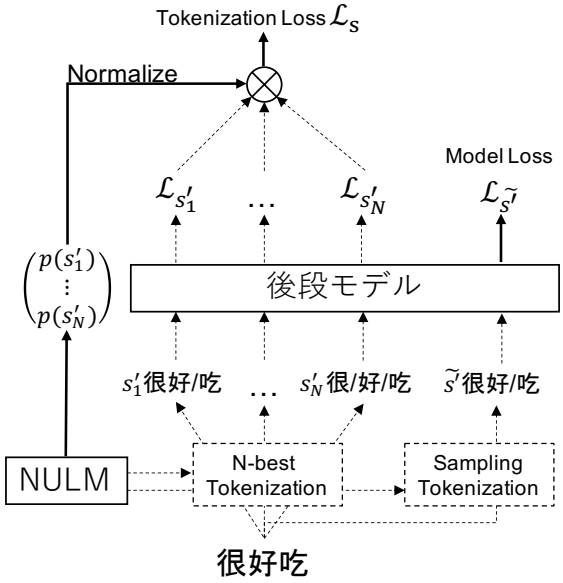

図 1: 提案手法による NULM と後段モデルの学習の概要．実線矢印は誤差逆伝播を行うパスを示す．

2 提案手法

提案手法の単語分割では，ニューラルユニグラム言語モデル (NULM) によって計算した単語のユニグラム確率 $p(w)$ を用いて，文 s を語彙 V に含まれる単語 w の系列 $s' = w_1, ..., w_I$ に変換する．$p(w)$ をもとに求めた N-best の単語分割候補 $s'_1, ..., s'_N$ を後段モデルに入力し，それぞれの出力に対応する損失値に単語分割の確率を掛け合わせることで損失値の重み付き和を計算する．この損失値に対して誤差逆伝播法を用いることで，損失値が低くなるような単語分割の確率が高くなるように NULM のパラメータを更新する．また，後段モデルの学習には NULM をもとに単語分割を一つサンプリングして用いることで，サブワード正則化による学習を行う．手法の概要を図 1 に示した．

2.1 N-best 単語分割による NULM の更新

提案手法の NULM では，以下のように単語分散表現 $\boldsymbol{v}_w$ を用いて単語のユニグラム確率を計算する．

$$d_w = \mathrm{MLP}(\boldsymbol{v}_w), \tag{1}$$

$$p(w) = \frac{\exp(d_w)}{\sum_{\hat{w}\in V}\exp(d_{\hat{w}})}. \tag{2}$$

入力文 s に含まれるあらゆる単語の確率 $p(w)$ に対して Forward-DP Backward-A*アルゴリズム [12] を適用し，入力文の N-best の単語分割候補を求める．さらに，それぞれの単語分割候補の確率 $p(s'_n) = \prod_{w\in s'_n} p(w)$ を用いて，各単語分割候補に対応する後段モデルの損失値 $\mathcal{L}_{s'_1}, ..., \mathcal{L}_{s'_N}$ を以下のように重み付ける．

$$\mathcal{L}_{s'_n} = q(f(s'_n), z), \tag{3}$$

$$a_n = \frac{p(s'_n)}{\sum_{m=1}^{N} p(s'_m)}, \tag{4}$$

$$\mathcal{L}_s = \sum_{n=1}^{N} a_n \mathcal{L}_{s'_n}. \tag{5}$$

ここで $f(\cdot)$ は単語分割済みの文を入力とし，後段タスクに応じた出力を計算する後段モデルである．また，$q(\cdot)$ は後段モデルの出力と教師信号 z を用いて損失値を計算する損失関数である．後段タスクが文書分類であれば，$f(\cdot)$ は各文書ラベルに対応する確率分布を出力し，$q(\cdot)$ は正解ラベルとの交差エントロピー損失を出力する．このような損失値の重み付き和に対して誤差逆伝播法を適用することで，その時点で損失値が最も低い単語分割の確率が向上するように NULM の更新が行われる．

式 (3) に用いる $q(f(\cdot),\cdot)$ は，単語の系列と教師信号を入力として損失値を出力するモデルであればよいので，提案手法は文書分類や機械翻訳などの様々な後段タスクに適用可能である．また，学習済みの NULM を用いてビタビアルゴリズム [13] による単語分割を行うことで，推論時には後段タスクに適切な単語分割を使用することができる．

2.2 後段モデルの更新

式 (5) で計算した $\mathcal{L}_s$ に対して単純に誤差逆伝播法を適用すると，後段モデルを計算グラフを繋いだ状態で N 個同時に計算することになり，メモリ使用量が莫大になる．そこで図 1 に示すように，$\mathcal{L}_s$ を用いた後段モデルのパラメータ更新は行わず，新たに単語分割 $\tilde{s}'$ をひとつサンプリングして後段モデルの学習に用いる．具体的には，$\mathcal{L}_{\tilde{s}'} = q(f(\tilde{s}'), z)$ に対して誤差逆伝播法を適用することで，後段モデルの更新を行う．これにより，提案手法では学習途中の NULM のパラメータを用いたサブワード正則化 [14, 15] による後段モデルの学習を行う．

単語分割 $\tilde{s}'$ を選択するために，NULM による単語のユニグラム確率を用いて $p(\tilde{s}')^\alpha / \sum_{k=1}^{K} p(s'_k)^\alpha$ からサンプリングを行う [14]．ここで α は分割の多様性を制御するハイパーパラメータであり，α が小さいほど一様な分布から単語分割をサンプリングする事になる．また，K はサンプリングの対象となる単語分割候補の数であり，$K = \infty$ の場合は，Forward Filtering Backward Sampling (FFBS) [16, 17] を用いたサンプリングを行う．

2.3 複数文の入力を用いた学習

機械翻訳タスクのソース側文とターゲット側文のように入力が複数となる場合には，一方の N-best 単語分割候補と他方のサプリングされた単語分割候補を組み合わせることで，NULM を更新するための損失値を計算する．つまり，複数入力のうち 1 つは N 個の候補を用い，その他はサンプリングした 1 つの候補を用いて計算する．具体的にソース側文 s とターゲット側文 t，それぞれの単語分割済み文を s', t' としたとき，$\mathcal{L}_{s'_n} = q(f(s'_n), \tilde{t}')$ と $\mathcal{L}_{t'_n} = q(f(\tilde{s}'), t'_n)$ によってソース側とターゲット側の NULM を更新する．また，後段モデルである機械翻訳モデルの学習には $\mathcal{L}_{\tilde{s}',\tilde{t}'} = q(f(\tilde{s}'), \tilde{t}')$ を用いる．

3 実験

3.1 文書分類

提案手法の有効性を確かめるために複数言語の文書分類タスクで実験を行い，その結果を表 1 に示した．Weibo(Zh)[1)], Twitter(Ja)[2)], Twitter(En)[3)] はそれぞれ中国語，日本語，英語の SNS コーパスを用いた感情分析タスクである．また，Genre，Rating タスクは中国語 [18][4)]，日本語 [19]，英語 [20][5)] の各言語の EC サイトのレビューデータから作成したジャンル予測とレート予測のタスクである．さらに複数文を

1) https://github.com/wansho/senti-weibo
2) http://www.db.info.gifu-u.ac.jp/data/Data_5d832973308d57446583ed9f
3) https://www.kaggle.com/c/twitter-sentiment-analysis2
4) http://yongfeng.me/dataset/
5) http://jmcauley.ucsd.edu/data/amazon/

表 1: 文書分類タスクでの実験結果 (F1 値).

	SentencePiece	OpTok	Ours
Weibo(Zh)	92.79	92.82	**93.06**
Twitter(Ja)	86.51	**86.97**	86.92
Twitter(En)	77.31	78.52	**78.88**
Genre(Zh)	47.95	48.18	**48.41**
Rating(Zh)	49.41	49.63	**49.76**
Genre(Ja)	49.84	50.15	**50.79**
Rating(Ja)	53.43	53.55	**53.69**
Genre(En)	71.68	**71.88**	71.83
Rating(En)	67.53	67.68	**67.90**
SNLI	76.75	77.04	**77.05**

入力とするタスクとして，SNLI データセット [21] での実験を行なった．

文書分類タスクでの実験設定は既存研究 [11] と揃え[6]，BiLSTM エンコーダーによる文書分類器をサブワード正則化を用いて学習した．単語分散表現は各コーパスの訓練データで事前学習し，文書分類の学習時には固定した．また，比較手法である OpTok と提案手法の語彙の初期化には，SentencePiece による同じ単語分割を使用し，単語分割の最適化についてのハイパーパラメータは $N = 3$ とした．

表 1 の実験結果より提案手法はほとんどのデータセットで OpTok を上回っており，既存手法に比べて性能が同等かそれ以上であることが確かめられた．この性能の差は，OpTok と提案手法における後段モデルの学習方法の違いに起因すると考えられる．OpTok は N-best の単語分割候補を用いて後段モデルを学習するが，推論時には 1-best の単語分割が後段モデルに入力されるため，学習と推論でギャップが生じてしまう．一方で提案手法による後段モデルの学習ではサンプリングされたひとつの単語分割候補のみを用いているため，学習と推論に差が生まれず性能の向上に繋がっていると考えられる．

3.2 機械翻訳

提案手法の機械翻訳タスクでの有効性を確かめるために IWSLT と WMT コーパスを用いた実験を行い，その結果を表 2 に示した．IWSLT コーパスでの実験には Transformer(small) [22] を使用し，全ての言語の語彙の規模を 16K として SentencePiece で単語分割を行なった．WMT コーパスでの実験では Transformer(base) を使用し，語彙の規模を 32K とした．サブワード正則化に関するハイパーパラメータは $k = \infty$，α は IWSLT:0.2，WMT:0.5 とした．また，提案手法で用いる単語分割の候補数 N は，IWSLT: 8，WMT: 3 とした．比較手法としてターゲット側言語の単語分割を最適化する DPE [10] を用いる．DPE による単語分割の学習には SentencePiece による単語分割を初期状態として用い，機械翻訳はソース側言語でサブワード正則化を用いた学習を行なった．また，提案手法の語彙の初期化にも SentencePiece による単語分割を使用した．コーパスは Moses[7]（中国語のみ jieba[8]）で事前に単語分割を施してから SentencePiece の学習を行った．機械翻訳の評価にはデトークナイズ後の BLEU4 値を用いた．

表 2: IWSLT(I) と WMT(W) コーパスを用いた機械翻訳タスクでの実験結果 (BLEU4)．エンコーダーとデコーダーで使用した単語分割手法をそれぞれ Enc, Dec として示した．また，SP は SentencePiece，R はサブワード正則化を示す．

	Enc Dec	SP SP	SP+R SP+R	SP+R DPE	Ours SP+R	SP+R Ours	Ours Ours
I14	De-En	33.79	35.03	35.02	34.90	**35.78**	35.13
	En-De	28.09	29.13	29.39	29.56	**29.57**	29.30
I15	Vi-En	28.70	28.78	28.85	29.34	**29.69**	29.44
	En-Vi	30.87	31.60	31.23	31.41	**31.74**	31.70
	Zh-En	20.44	21.17	21.38	21.63	21.65	**21.89**
	En-Zh	14.40	15.25	15.21	15.45	**15.59**	15.31
I17	Ar-En	29.23	29.39	29.37	29.48	**30.04**	29.78
	En-Ar	15.45	17.75	17.83	**18.49**	18.18	18.21
	Fr-En	37.87	38.43	38.52	**38.82**	38.68	38.58
	En-Fr	37.95	39.83	39.90	40.01	**40.08**	39.68
W09	Hu-En	14.84	15.51	**15.75**	15.73	15.74	15.60
	En-Hu	11.02	12.14	12.30	12.30	**12.37**	12.33
W14	De-En	31.46	31.89	31.97	**32.19**	31.98	31.90
	En-De	27.10	27.41	27.49	**27.62**	27.52	27.44
W16	Ro-En	29.10	31.79	31.67	31.80	**31.83**	31.72
	En-Ro	21.78	24.05	24.29	24.36	**24.53**	24.03

表 2 の結果より，多くの言語対での実験において提案手法を用いた学習の性能が既存手法を上回ることが確認され，提案手法の有効性が示された．特に，デコーダーのみに提案手法を適用した設定が多くの場合でもっとも性能が高く，ほとんどの言語でエンコーダーとデコーダーの双方に提案手法を用いたときの性能が低くなっている．ここから，ソース側言語とターゲット側言語の単語分割の最適化を同時に行うことが難しく，性能の低下につながっていると考えられる．両側の単語分割を同時に最適化する方策についての追加実験を付録 A に掲載した．

6) 実験に使用したデータセットの詳細と，単語分散表現，SentencePiece の学習済みモデルを以下で公開している．https://github.com/tatHi/optok

7) https://github.com/moses-smt/mosesdecoder

8) https://github.com/fxsjy/jieba

4 分析

機械翻訳タスクで得られた提案手法の単語分割について分析を行う．英中対訳ペアのうち中国語側をSentencePiece，英語側を提案手法で学習し，提案手法によって得られた単語分割を表 3 に示した．

提案手法によるソース側言語の単語分割 (表 3a) では，“hav-e” や “hour-s” のようにトークンの形態素や接尾辞を細かく分割していることが確認される．また，ターゲット側の単語分割（表 3b）では既存手法の DPE と同様に，動詞の接尾辞 “-ed” を分割する傾向が見られる．一方で DPE や提案手法によるソース側言語の単語分割とは異なり，ターゲット側言語を細かく分割する傾向は見られなかった．

IWSLT コーパスにおいて各手法で得られた単語分割と，初期状態である SentencePiece の単語分割との学習データにおけるトークン数の割合を表 4 に示した．表において，1 を超える値は SentencePiece に比べてトークン数が増加していることを表す．結果より，提案手法でソース側言語の単語分割を学習した場合は初期状態に比べてトークン数が増大しており，提案手法がソース側言語を細かい単位で分割していることがわかる．これは，エンコーダーのニューラルネットワークの表現力が豊かであるために，細かい単位で入力しても機械翻訳の性能が低下しないことを示唆する．この傾向は，エンコーダーの入力を文字レベルにすることで性能向上が得られるという既存研究の結果と合致する [23, 24]．

また，英中翻訳以外の全ての言語対で，提案手法によるターゲット側の単語分割は初期状態と比べてトークン数がわずかに少なくなっている．これは，提案手法が言語モデルベースの SentencePiece による単語分割の粒度を維持しつつ，デコードしやすい単位に分割し直した結果であると考えられる．英中翻訳ではターゲット側であっても，トークン数が大きく増加している．これは中国語の 1 文字が持つ情報が他の言語に比べて多く，エンコーダー側の単語の粒度と合わせるために細かく分割したと考えられる．比較手法である DPE は言語対ごとにターゲット側のトークン数が変化しており，提案手法との差が見られる．DPE はソース側の単語分割を用いてターゲット側の単語分割を決定するが，提案手法はターゲット側の学習済みユニグラム言語モデルのみを用いて単語分割を行うために単語分割の自由度が低く，このような差が生まれていると考えられる．

表 3: 英中ペアにて英語側の単語分割を最適化した時の SentencePiece(SP) と DPE，提案手法の比較．

(a) 英中翻訳のソース言語の単語分割

SP	Student s **don ’ t have** long **hours** of learning .
OURS	Student s **do n ’ t hav e** long **hour s** of learning .
TGT	学生 在 校 学 习 时间 不 长 。

(b) 中英翻訳のターゲット言語の単語分割

SRC	引力 与 其它 力 分 隔 开来
SP	Gra vity **separate d away** from the other force s .
DPE	Gra vity **separat ed a way** from the other force s .
OURS	Gra vity **separat ed away** from the other force s .

表 4: 学習データ全体で，各手法で学習された単語分割のトークン数の初期状態に対する割合．初期状態の単語分割は SentencePiece によるものである．

	Enc Dec	Ours SentencePiece	SentencePiece Ours	SentencePiece DPE
I14	De-En	2.5353	0.9992	1.0439
	En-De	1.3809	0.9996	0.9923
I15	Vi-En	1.5320	0.9993	1.0428
	En-Vi	1.4650	0.9999	0.9923
	Zh-En	1.5175	0.9994	0.9907
	En-Zh	1.3516	1.4713	1.0346
I17	Ar-En	2.5350	0.9997	0.9952
	En-Ar	1.4765	0.9994	0.9945
	Fr-En	1.7194	0.9996	1.0001
	En-Fr	1.5996	0.9997	0.9935

これらの分析から，提案手法は言語や後段モデルなどの後段タスクの情報に応じて異なる単語分割を学習していることが示唆される．

5 おわりに

本稿では，単語分割をタスクに応じて最適化する手法を提案した．提案手法では単語分割器を後段モデルの学習に組み込むことで，タスクやモデルに応じて単語分割を更新する．実験結果より，提案手法は文書分類と機械翻訳の複数データセットで既存手法を上回る性能であり，その有効性を確認した．提案手法は損失値を用いて単語分割の更新を行うため，応用先はこれらのタスクに限らない．今後は文法誤り訂正やスタイル変換などのタスクに提案手法が有効であるかを検証する．

謝辞 本研究成果は，国立研究開発法人情報通信研究機構（NICT）の委託研究「多言語音声翻訳高度化のためのディープラーニング技術の研究開発」により得られたものです．

参考文献

[1]Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pp. 177–180. Association for Computational Linguistics, 2007.

[2]Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. http://taku910.github.io/mecab/, 2006.

[3]Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, pp. P1715–1725, 2016.

[4]Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.

[5]Jia Xu, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. Bayesian semi-supervised chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 1017–1024. Association for Computational Linguistics, 2008.

[6]Pi-Chuan Chang, Michel Galley, and Christopher D Manning. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pp. 224–232, 2008.

[7]ThuyLinh Nguyen, Stephan Vogel, and Noah A Smith. Nonparametric word segmentation for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 815–823. Association for Computational Linguistics, 2010.

[8]Miguel Domingo, Mercedes Garcıa-Martınez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. How much does tokenization affect neural machine translation? *arXiv preprint arXiv:1812.08621*, 2018.

[9]Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. Optimizing segmentation granularity for neural machine translation. *Machine Translation*, pp. 1–19, 2020.

[10]Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3042–3051, Online, July 2020. Association for Computational Linguistics.

[11]Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Optimizing word segmentation for downstream task. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1341–1351, Online, November 2020. Association for Computational Linguistics.

[12]Masaaki Nagata. A stochastic japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pp. 201–207. Association for Computational Linguistics, 1994.

[13]Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, Vol. 13, No. 2, pp. 260–269, 1967.

[14]Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2018.

[15]Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1882–1892, Online, July 2020. Association for Computational Linguistics.

[16]Steven L Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, Vol. 97, No. 457, pp. 337–351, 2002.

[17]Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 100–108. Association for Computational Linguistics, 2009.

[18]Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proceedings of the 24th international conference on world wide web*, pp. 1373–1383, 2015.

[19]Rakuten Inc. Rakuten dataset. Informatics Research Data Repository, National Institute of informatics. (dataset)., 2014.

[20]Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pp. 507–517, 2016.

[21]Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, 2015.

[22]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, Vol. 30, pp. 5998–6008, 2017.

[23]Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[24]Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 365–378, 2017.

A　複数入力に対する同時最適化

3.2 節では，エンコーダー側とデコーダー側の両方の単語分割を提案手法で同時に最適化することが性能の低下に繋がると説明した．これはエンコーダー側とターゲット側で単語分割の方策に差があり (4 章)，単語分割の最適化が安定しないためだと考えられる．そこで，双方の単語分割の最適化を段階的に行うことで，両方の単語分割を最適化しつつ性能の低下を防ぐ方法を考える．本研究の実験における 100 エポックの学習の中で，1) エンコーダー側の単語分割のみを前半 50 エポックで学習し，後半はエンコーダー側の単語分割を固定してデコーダー側の単語分割のみを学習する，2) 前半にデコーダー側の単語分割を学習し，後半でエンコーダー側を学習する，3) 各ミニバッチ学習ごとにランダムにエンコーダー側とデコーダー側を選択して片方のみの単語分割を学習する，の 3 つの方策を実験する．

表 5 に示した実験結果より，1) エンコーダー側の単語分割を先に学習し，その後デコーダー側の単語分割を学習する方策がもっとも高い性能であることが分かった．特に Vi-En，En-Vi，Zh-En では表 2 の数値を含めても最高性能に達しており，方策 1 が優れていることがわかる．また，表 4 で示したようにエンコーダー側の単語分割は粒度が細かくなる傾向があるため，学習の後半で分割が大きく変わる方策 2 では性能が低下することが分かった．

表 5: 両側の単語分割を同時に最適化するための各方策による，IWSLT15 での性能の差（BLEU4）．Both は表 2 の Ours-Ours の値を引用．

	Both	1) EncDec	2) DecEnc	3) Random
Vi-En	29.44	**30.22**	29.47	29.37
En-Vi	31.70	**31.78**	31.33	31.70
Zh-En	21.89	**21.99**	21.82	21.66
En-Zh	15.31	**15.54**	14.88	15.14

B　ハイパーパラメータの影響

提案手法では N 個の単語分割候補を用いて NULM の最適化を行う．このハイパーパラメータ N について，文書分類タスクと機械翻訳タスクの性能に与える影響をそれぞれ図 2 と図 3 に示した．図では，3 章での実験に用いた N の値で得られた後段モデルの性能との差を示す．

文書分類タスクでの実験では，Twitter と Weibo の感情分析タスクを用いた性能の差を検証した．実験より，文書分類タスクでは提案手法の N の値による大きな性能の差は見られなかった．同様の実験を提案手法がベースとしている OpTok [11] で行った結果と比較すると，提案手法はハイパーパラメータ N が大きくなっても，性能が落ちないことが確認される．OpTok が N 個の単語分割候補に対応する文ベクトルの重み付き和で後段モデルの学習を行うのに対して，提案手法は N の値にかかわらず 1 つの単語分割候補のみを用いて後段モデルの学習を行うために，N の影響を受けにくいと考えられる．

機械翻訳タスクでの実験では，IWSLT15 の越英翻訳ペアを用いて N の影響を確認した．結果より，ターゲット側言語のみに提案手法を適用した場合は N の値による性能の大きな差はなく，ソース側言語のみに提案手法を用いた場合は N を大きく取ることで性能が向上していることが確認された．これは 4 章での分析で述べたように，エンコーダーの表現力の高さのために初期状態から離れた単語分割候補を用いることが性能向上に繋がり得ることに起因すると考えられる．一方で，ソース側言語とターゲット側言語の双方に提案手法を適用する場合は N を大きくすることで若干の性能低下が見られる．性能低下の程度は小さいが，両側の言語の単語分割を同時に更新する場合は N が大きくなることで学習が不安定になると考えられる．

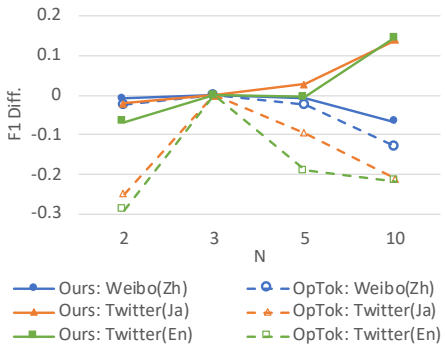

図 2: N による文書分類タスクの性能の差

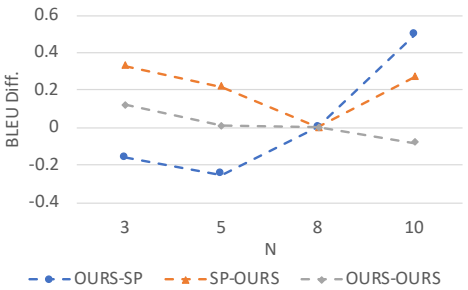

図 3: N による機械翻訳タスクの性能の差