近傍文検索を用いたサブセット kNN ニューラル機械翻訳

出口 祥之 [1,2]　渡辺 太郎 [1]　松井 勇佑 [3]　内山 将夫 [2]　田中 英輝 [2]　隅田 英一郎 [2]
[1] 奈良先端科学技術大学院大学　[2] 情報通信研究機構　[3] 東京大学
[1]{deguchi.hiroyuki.db0,taro}@is.naist.jp [3]matsui@hal.t.u-tokyo.ac.jp
[2]{mutiyama,hideki.tanaka,eiichiro.sumita}@nict.go.jp

概要

kNN-MT [1] は，翻訳時に用例検索を組み込むことで，モデルを追加学習することなくニューラル機械翻訳（NMT）の精度を改善する．しかし，翻訳中の各時刻で，対訳データの全目的言語トークンに対して近傍探索を行うため，翻訳速度が通常の NMT の 100〜1000 倍ほど遅くなるという問題点がある．本研究では検索対象を入力文の近傍事例に絞ることで kNN-MT の高速化を図る．また，ルックアップテーブルを用いた効率的な距離計算により，さらなる高速化を目指す．複数の翻訳実験を行ったところ，従来法より最大で 1.6 BLEU ポイント精度が改善し，最大 132.2 倍速度が改善することを確認した．

1　はじめに

Transformer [2] ニューラル機械翻訳（Neural Machine Translation; NMT）は，従来の統計的手法より翻訳精度と流暢性が高く，注目を浴びている．近年，訓練データと異なるドメインの翻訳精度が低下する課題に対処するため，Transformer NMT に用例検索の手法を組み込んだ kNN-MT [1] が提案されている．kNN-MT は，訓練済み NMT モデルの中間表現を用いて，データストアと呼ばれるキー・値メモリに対訳データを格納し，翻訳中の各時刻で k 近傍事例を探索する．これにより，モデルを追加学習することなく翻訳精度を改善できることが報告されている．しかし，従来の kNN-MT では対訳データの目的言語側の全トークンを対象に近傍探索するため，翻訳速度が通常の Transformer NMT より 100〜1000 倍ほど遅くなるという問題点がある．

本研究では入力文の近傍事例のみに検索対象を絞ることで kNN-MT の高速化を図る．また，ルックアップテーブルを用いて各事例との距離を効率的に求めることで，さらなる高速化を目指す．WMT19 独英翻訳とドメイン適応翻訳実験を行い，提案法は従来法より翻訳精度が最大で 1.6 BLEU ポイント，翻訳速度が最大で 132.2 倍改善することを確認した．

2　kNN-MT

データストア構築　一般的な NMT は，原言語文 $\boldsymbol{x} = (x_1, x_2, \ldots, x_{|\boldsymbol{x}|})^\top \in \mathcal{V}_X^{|\boldsymbol{x}|}$ が与えられたとき目的言語文 $\boldsymbol{y} = (y_1, y_2, \ldots, y_{|\boldsymbol{y}|})^\top \in \mathcal{V}_Y^{|\boldsymbol{y}|}$ を先頭から順に生成する．ただし，$|\boldsymbol{x}|, |\boldsymbol{y}|$ はそれぞれ文 $\boldsymbol{x}, \boldsymbol{y}$ の長さ，$\mathcal{V}_X, \mathcal{V}_Y$ はそれぞれ入力語彙と出力語彙を表す．時刻 t に出力されるトークン y_t は，原言語文 $\boldsymbol{x}$ と時刻 t までに生成した目的言語トークン系列 $\boldsymbol{y}_{<t}$ から計算される確率 $p(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t})$ に基づき生成される．kNN-MT は，翻訳前に予め，データストアと呼ばれる D 次元ベクトルと語彙の組からなるキー・値メモリ $\mathcal{M} \subseteq \mathbb{R}^D \times \mathcal{V}_Y$ を構築する．キーは NMT モデルに teacher forcing 方式 [3] で対訳文対 $(\boldsymbol{x}, \boldsymbol{y})$ を入力したときに得られるデコーダ最終層の中間表現，値はキーベクトルが得られたときに出力されるべき正解の目的言語トークン $y_t \in \mathcal{V}_Y$ であり，以下のような式で定義される．

$$\mathcal{M} = \{(f(\boldsymbol{x}, \boldsymbol{y}_{<t}), y_t) \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}, 1 \leq t \leq |\boldsymbol{y}|\}. \quad (1)$$

なお，$\mathcal{D}$ は対訳データ，$f : \mathcal{V}_X^{|\boldsymbol{x}|} \times \mathcal{V}_Y^{t-1} \to \mathbb{R}^D$ は原言語文と生成済みトークンからデコーダ最終層の D 次元中間表現ベクトルを得るような NMT モデルを表す関数とする．本研究では順伝播層への入力ベクトルをキーとする．

翻訳時　出力トークン $y_t \in \mathcal{V}_Y$ の生成確率は kNN 確率と MT 確率の線形補間により計算される．

$$P(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \lambda p_{\mathrm{kNN}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) + (1-\lambda) p_{\mathrm{MT}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}), \quad (2)$$

なお，λ はそれぞれの確率に対する重み付けのハイパーパラメータである．時刻 t におけるクエリベクトルを $f(\boldsymbol{x}, \boldsymbol{y}_{<t})$，データストア $\mathcal{M}$ に対する k 近傍の上位 i 番目のキー・値をそれぞれ $\boldsymbol{k}_i \in \mathbb{R}^D, v_i \in \mathcal{V}_Y$

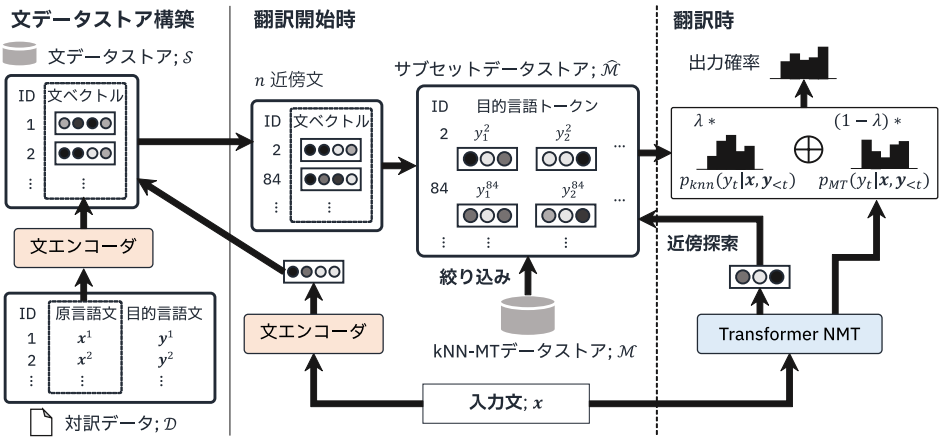

図 1　サブセット kNN-MT

とし，τ を kNN 確率の温度パラメータとすると，p_{kNN} は次式のように計算される．

$$p_{\mathrm{kNN}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) \propto \sum_{i=1}^{k} \mathbb{1}_{y_t=v_i} \exp\left(\frac{-\|\boldsymbol{k}_i - f(\boldsymbol{x}, \boldsymbol{y}_{<t})\|_2^2}{\tau}\right), \tag{3}$$

3　提案法: サブセット kNN-MT

提案法（図 1）は入力文の情報を活用することで翻訳開始時に探索対象を削減する（3.1 節）．また，翻訳中の各時刻で，絞り込んだ事例の中から上位 k 近傍を求める際，効率的な計算手法を用いることでクエリとキーの間の距離を高速に求める（3.2 節）．

3.1　サブセット探索

文データストア構築　提案法では文データストア $\mathcal{S}$ を構築して対訳データの原言語文の文ベクトルをキー，目的言語文を値として格納する．

$$\mathcal{S} = \{(h(\boldsymbol{x}), \boldsymbol{y}) \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}\}, \tag{4}$$

ただし，$h : \mathcal{V}_X^{|\boldsymbol{x}|} \to \mathbb{R}^{D'}$ は原言語文の D' 次元文ベクトルを得る文エンコーダモデルを表す関数とする．

翻訳時　翻訳開始時に，入力文の文ベクトルとのユークリッド距離が近い n 近傍文を文データストア $\mathcal{S}$ から探索し，n 近傍文集合 $\hat{\mathcal{S}} \subset \mathcal{S}$ を得る．ここで，kNN-MT の検索対象を次式のように絞り込む．

$$\hat{\mathcal{M}} = \{(f(\boldsymbol{x}, \boldsymbol{y}_{<t}), y_t) \mid (h(\boldsymbol{x}), \boldsymbol{y}) \in \hat{\mathcal{S}}, 1 \le t \le |\boldsymbol{y}|\}, \tag{5}$$

なお，$\hat{\mathcal{M}} \subset \mathcal{M}$ は近傍文の事例に絞り込まれたデータストアである．翻訳中は $\hat{\mathcal{M}}$ から近傍探索する以外は従来法と同様である．すなわち，提案法は検索対象とする文を $|\mathcal{D}|$ 文から $n(\ll |\mathcal{D}|)$ 文に事前に絞り込み，その文に対し出力の k 近傍を探索する．

3.2　ルックアップによる効率的な距離計算

直積量子化　データストアは対訳データ中の全目的言語トークンの中間表現を保持するため，メモリ上に直接ロードすることは困難[1)]である．本研究では，従来の kNN-MT と同様に，直積量子化（Product Quantization; PQ）[4] と呼ばれるベクトル量子化手法を用いてデータストアを圧縮する．PQ は D 次元ベクトルを $\frac{D}{M}$ 次元ずつ M 個のサブベクトルに分割し，それぞれのサブベクトルを量子化する．量子化のためのコードブックは各 $\frac{D}{M}$ 次元部分空間で学習され，m 番目の空間のコードブック $\mathcal{C}^m$ は以下のように表される．

$$\mathcal{C}^m = \{\boldsymbol{c}_1^m, \ldots, \boldsymbol{c}_L^m\},\ \boldsymbol{c}_l^m \in \mathbb{R}^{\frac{D}{M}}. \tag{6}$$

なお，本研究では各コードブックの大きさを $L = 256$ に設定する．ベクトル $\boldsymbol{q} \in \mathbb{R}^D$ は以下のようにしてコードベクトル $\bar{\boldsymbol{q}}$ に量子化される．

$$\bar{\boldsymbol{q}} = [\bar{q}^1, \ldots, \bar{q}^M]^\top \in \{1, \ldots, L\}^M, \tag{7}$$

$$\bar{q}^m = \operatorname*{argmin}_l \|\boldsymbol{q}^m - \boldsymbol{c}_l^m\|_2^2,\ \boldsymbol{q}^m \in \mathbb{R}^{\frac{D}{M}}. \tag{8}$$

量子化コード上での距離計算　提案法では，PQ によって量子化されているサブセットデータストア $\mathcal{M}_n$ に対して k 近傍を探索する際，以下に示す Asymmetric Distance Computation (ADC) [4] を用いて効率的にベクトル間距離を計算する．手法の概要を図 2 に示す．

ADC は，クエリベクトル $\boldsymbol{q} \in \mathbb{R}^D$ と，N 個の M 次元キーベクトル $\bar{\mathcal{K}} = \{\bar{\boldsymbol{k}}_1, \ldots, \bar{\boldsymbol{k}}_N\}$ のそれぞれとの距

1)　4.1 節の実験では 862,648,422 トークン分の 1024 次元ベクトルを使用する．このとき，データストア容量は約 3.2 TiB ≃ 862,648,422 トークン × 1024 次元 × 32 bit float/8 bit/1024[4].

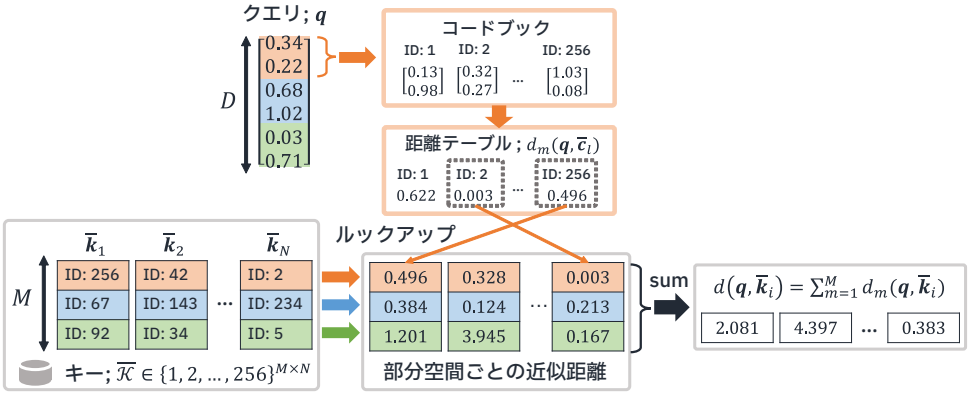

図 2　ルックアップテーブルを用いた近似距離計算

離を以下のようにして求める．まず，各部分空間 m において，$\boldsymbol{q}^m$ とコード $\boldsymbol{c}_l^m \in \mathcal{C}^m$ との距離テーブル $\boldsymbol{A}^m \in \mathbb{R}^L$ を求める．

$$A_l^m = \|\boldsymbol{q}^m - \boldsymbol{c}_l^m\|_2^2. \tag{9}$$

このとき，クエリと各キーとの距離 $d(\boldsymbol{q}, \bar{\boldsymbol{k}}_i)$ は次式により求まる．

$$d(\boldsymbol{q}, \bar{\boldsymbol{k}}_i) = \sum_{m=1}^{M} d_m(\boldsymbol{q}^m, \bar{\boldsymbol{k}}_i^m), \tag{10}$$

$$\text{where } d_m(\boldsymbol{q}^m, l) = A_l^m. \tag{11}$$

ADC を用いるとキー集合 $\overline{\mathcal{K}}$ を復号しないため，量子化された $\hat{\mathcal{M}}$ から効率的に k 近傍を探索できる．

4　実験

従来法と提案法の翻訳精度と翻訳速度を比較するため，翻訳実験を行った．翻訳精度は sacreBLEU [5] で評価した．全ての実験において，NVIDIA V100 GPU を 1 基使用した．NMT モデルには FAIRSEQ [6] の WMT19 独英訓練済み Transformer big モデルを用いた．翻訳時のビーム幅は 5，文長正規化パラメータは 1.0，バッチサイズは 12,000 トークンとした．kNN-MT の探索近傍数は $k = 16$，kNN 確率の温度は $\tau = 100$ に設定した．従来法の近傍探索と提案法の近傍文探索には FAISS [7] を使用し，PQ のサブベクトル数は $M = 64$ に設定した．提案法のサブセットからの k 近傍探索（ADC 含む）は PyTorch で実装した．提案法で用いる文エンコーダには *LaBSE* [8] と Transformer NMT のエンコーダ最終層の中間表現の平均 *AvgEnc* を使用し，性能を比較した．

4.1　WMT19 独英翻訳

翻訳速度と翻訳精度を WMT19 独英翻訳（newstest2019; 2,000 文）で評価し，従来法および先行研究の fast kNN-MT [9] と提案法の性能を比較した．データストアには WMT19 独英翻訳の対訳データのサブワード化後の文長が 250 以下かつ対訳文の文長比が 1.5 以内のデータのみを用い，29,540,337 文から得られた 862,648,422 トークンから構築した．近傍文探索による絞り込み数は $n = 512$ とした．

表 1　WMT19 独英翻訳の精度と速度の比較．“h: ” は文エンコーダモデルを示し，“–ADC” はルックアップテーブルを用いなかったときの結果を示す．

モデル	↑BLEU%	↓秒	↑トークン/秒
ベース MT	39.2	7.5	6375.2
kNN-MT	40.1	2446.0	19.6
fast kNN-MT [9]	40.3	162.7	286.9
提案法 (h: LaBSE)	40.1	21.9	2191.4
– ADC		107.7	444.8
提案法 (h: AvgEnc)	39.9	26.4	1816.8
– ADC		131.9	364.2

実験結果を表 1 に示す．表より，kNN-MT を用いることで，ベース MT より精度が 0.9 ポイント改善するものの，速度は 326.1 倍遅くなる．一方，提案法（LaBSE）を用いることで，精度が低下することなく kNN-MT より 111.8 倍翻訳速度が改善している．また，AvgEnc は kNN-MT より 0.2 ポイント精度が低下したが，追加のモデルを使用することなく翻訳速度が 92.7 倍ほど改善している．表中の “ – ADC” は距離計算に ADC を用いなかった場合を示しており，LaBSE，AvgEnc ともに約 5 倍程度速度が低下することが確認できる．

4.2　ドメイン適応翻訳

IT, Koran, Law, Medical, Subtitles の 5 つのドメインの翻訳実験 [10, 11] を行った．NMT モデルには 4.1

表 2 ドメイン適応翻訳の精度（BLEU%）と速度（トークン/秒）の比較．**太字**は各ドメインにおける最高精度を表す．

モデル	IT		Koran		Law		Medical		Subtitles	
	↑精度	↑速度	↑精度	↑速度	↑精度	↑速度	↑精度	↑速度	↑精度	↑速度
ベース MT	38.7	4433.2	17.1	5295.0	46.1	4294.0	42.1	4392.1	29.4	6310.5
kNN-MT	41.0	22.3	19.5	19.3	52.6	18.6	48.2	19.8	29.6	30.3
提案法 (*h*: LaBSE)	**41.9**	2362.2	**20.1**	2551.3	**53.6**	2258.0	**49.8**	2328.3	29.9	3058.4
提案法 (*h*: AvgEnc)	**41.9**	2197.8	19.9	2318.4	53.2	1878.8	49.2	2059.9	**30.0**	3113.0

表 3 Medical ドメインにおける実際の翻訳例．

入力文	Jede Filmtablette enthält 7,5 mg Olanzapin Sonstiger Bestandteil:
参照訳	Each **film-coated** tablet contains 7.5 mg olanzapine.
ベース MT	Each film tablet contains 7.5 mg of olanzapine.
kNN-MT	Each film tablet contains 7.5 mg olanzapine.
提案法	Each **film-coated** tablet contains 7.5 mg olanzapine.

表 4 表 3 の提案法における上位 3 近傍文の探索結果．

入力文	Jede Filmtablette enthält 7,5 mg Olanzapin Sonstiger Bestandteil:
S-1	Jede Tablette enthält 7,5 mg Olanzapin
S-2	Jede Filmtablette enthält 5 mg Olanzapin Sonstiger Bestandteil:
S-3	Jede Filmtablette enthält 2,5 mg Olanzapin Sonstiger Bestandteil:
T-1	Each coated tablet contains 7.5 mg olanzapine
T-2	Each **film-coated** tablet contains 5 mg olanzapine.
T-3	Each **film-coated** tablet contains 2.5 mg olanzapine.

節と同様のモデルを用いた．本実験では，入力文のドメインが未知のオープンドメイン設定を想定するため，データストアの対訳データには 4.1 節で用いたデータと各対象ドメインの対訳データを全てを結合したデータを用い，30,843,860 文から得られた 895,891,420 トークンを使用した．近傍文探索による絞り込み数は $n = 256$ とした．

実験結果を表 2 に示す．全てのドメインにおいて，kNN-MT を用いることで，ベース MT と比較して翻訳精度は改善しているが，速度は約 200 倍以上遅くなっている．一方で，提案法を用いることで，従来法より最大で 132.2 倍翻訳速度が改善されている．さらに，翻訳速度だけでなく，翻訳精度が kNN-MT よりも最大 1.6 ポイント改善している．

Medical ドメインにおける実際の翻訳例と近傍事例の検索結果をそれぞれ表 3, 4 に示す．提案法は “*h*: LaBSE” の結果を示している．表 3 より，提案法では，Medical ドメインの定訳である “Filmtablette” → “film-coated tablet” を正しく翻訳できていることが確認できる．このときの “入力文の上位 3 近傍文 (S-{1,2,3})” と，“それらの対訳文 (T-{1,2,3})” を表 4 に示す．表 4 より，検索対象であるサブセット内に “film-coated” が含まれている．提案法では近傍事例のみからなるサブセットを探索することで，より適切な単語を訳出できるようになったと考えられる．

5 関連研究

用例を用いた機械翻訳手法は，類推に基づく機械翻訳 [12] によって提案され，NMT に対する拡張として，編集距離に基づいて用例を検索する手法が提案されている [13, 14]．

kNN-MT の速度を改善した fast kNN-MT [9] が提案されている．fast kNN-MT は入力文中の各語彙の近傍事例とその単語アライメントをもとに検索対象を絞り込む．機械翻訳以外では，言語モデルに近傍探索手法を用いた kNN-LM [15] や，kNN-LM を高速化した Efficient kNN-LM [16] が提案されている．

ベクトル近傍探索分野では Reconfigurable Inverted Index (Rii) [17] が提案されている．従来の探索法は全探索のみを想定しているが，Rii では動的に作成されるサブセットに対する探索を可能とする．

6 おわりに

本研究では，kNN-MT の翻訳速度を改善した．提案法では，入力文の近傍探索により kNN-MT の検索対象を絞り込み，ルックアップテーブルを用いて効率的にベクトル間距離を計算する．複数の翻訳実験の結果，提案法を用いることで kNN-MT より最大 1.6 BLEU ポイントの精度改善と，最大 132.2 倍の速度改善を確認した．今後は，機械翻訳以外のタスクへの応用も検討していきたい．

謝辞

本研究の一部は JSPS 科研費 22J11279 の助成を受けたものである．ここに謝意を表する．

参考文献

[1] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In **International Conference on Learning Representations (ICLR)**, 2021.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, **Advances in Neural Information Processing Systems 30**, pp. 5998–6008. Curran Associates, Inc., 2017.

[3] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. **Neural Computation**, Vol. 1, No. 2, pp. 270–280, 1989.

[4] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 33, No. 1, pp. 117–128, 2011.

[5] Matt Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[6] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)**, pp. 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2019.

[8] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[9] Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. Fast nearest neighbor machine translation. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 555–565, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[10] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In **Proceedings of the First Workshop on Neural Machine Translation**, pp. 28–39, Vancouver, August 2017. Association for Computational Linguistics.

[11] Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7747–7763, Online, July 2020. Association for Computational Linguistics.

[12] Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In **Proc. of the International NATO Symposium on Artificial and Human Intelligence**, pp. 173–180, 1984.

[13] Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. Guiding neural machine translation with retrieved translation pieces. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1325–1335, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] J Gu, Y Wang, K Cho, and V O K Li. Search engine guided neural machine translation. **AAAI**, 2018.

[15] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In **International Conference on Learning Representations**, 2020.

[16] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 5703–5714, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[17] Yusuke Matsui, Ryota Hinami, and Shin'ichi Satoh. Reconfigurable inverted index. In **ACM International Conference on Multimedia (ACMMM)**, pp. 1715–1723, 2018.

[18] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 36, No. 4, pp. 744–755, 2014.

[19] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. **IEEE transactions on pattern analysis and machine intelligence**, Vol. 42, No. 4, pp. 824–836, 2018.

表 5　WMT19 独英翻訳においてバッチサイズを変えたときの翻訳速度．

		B_∞		B_1	
モデル	↑BLEU%	↓秒	↑トークン/秒	↓秒	↑トークン/秒
Base MT	39.2	7.5	6375.2	371.5	129.14
kNN-MT	40.1	2446.0	19.6	18928.7	2.5
fast kNN-MT [9]	40.3	162.7	286.9	1725.1	27.1
提案法 (h: LaBSE)	40.1	21.9	2191.4	404.6	118.4
提案法 (h: AvgEnc)	39.9	26.4	1816.8	493.7	97.3

表 6　データストアのインデックス設定．

設定項目	従来法 kNN-MT データストア	提案法 文データストア	kNN-MT データストア
PQ サブベクトル数	$M = 64$	$M = 64$	$M = 64$
ベクトル回転	Optimized PQ [18]	Optimized PQ [18]	—
次元削減	—	—	PCA: 1024 次元→ 256 次元
転置インデックス	131,072 クラスタ	32,768 クラスタ	—
探索クラスタ数	近傍 64 クラスタ	近傍 64 クラスタ	—
粗量子化器	HNSW Flat [19] (エッジ数: 32)	Flat	—

A　モデル，データセット

モデルとデータセットの URL を以下に示す．

- NMT モデル: https://dl.fbaipublicfiles.com/fairseq/models/wmt19.de-en.ffn8192.tar.gz
- ドメイン適応データ: https://github.com/roeeaharoni/unsupervised-domain-clusters

B　実験に使用した計算機

実験に使用した計算機のスペックは以下の通りである．

- CPU: Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz (18 コア × 2 基)
- Memory: 768 GB
- GPU: NVIDIA Tesla V100 (1 基使用)
- OS: CentOS 7.6

C　バッチサイズと翻訳速度

本文中の実験ではバッチサイズを 12,000 トークンに設定して速度を比較した．表 5 は，翻訳時のバッチサイズについて，文書翻訳等を想定した 12,000 トークン（B_∞）とリアルタイム翻訳等を想定した 1 文（B_1）としたときの翻訳速度の比較を示す．

D　kNN インデックスの詳細

近傍探索インデックスの詳細な設定を表 6 に示す．