知識蒸留によるニューラル機械翻訳の高速化検討

今村 賢治
国立研究開発法人 情報通信研究機構
kenji.imamura@nict.go.jp

1 はじめに

知識蒸留 (knowledge distillation)[1, 2] は、teacher-student 方式とも呼ばれ、教師モデルが学習した知識を生徒モデルに移行する方法の一つである。教師モデルと生徒モデルは、アーキテクチャを含めて異なるものを用いることができる。

一方、ニューラル機械翻訳 (NMT) においても、実行時の高速化は、実用性を高めるために必須である。高速化方法にはいくつもの種類があるが、シンプルな方法としては、モデルを縮小（パラメーターの削減）することによる計算量削減と、ビーム幅を狭くすることによる探索空間の削減がある。知識蒸留と組み合わせる場合、教師モデルより小規模な生徒モデルにすることで、高速化する場合が多い。

本稿では、知識蒸留と組み合わせた機械翻訳の高速化について、翻訳品質と翻訳速度の観点から、実験的に検討を行う。知識蒸留と組み合わせた場合、どの方法が翻訳品質を落とさずに高速化が可能であるのか、検証するのが本稿の目的である。

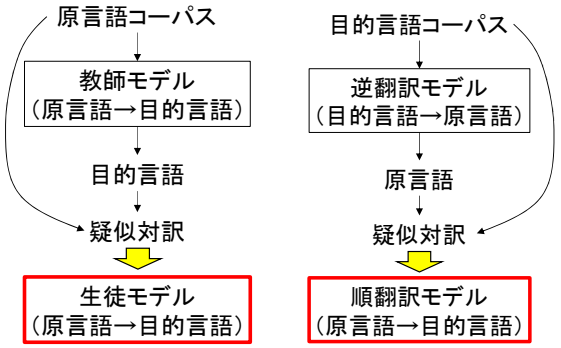
2 知識蒸留と高速化

2.1 知識蒸留

知識蒸留自体は、教師モデル（teacher model、親モデル parent model とも呼ばれる）が出力する結果を模倣するように、生徒モデル (student model) を学習することを言う。

たとえば、マルチクラス分類タスクでは、出力は全クラスの事後確率分布であるので、（正解クラスだけでなく）すべてのクラスの確率分布が教師モデルと一致するように損失関数を設定し、生徒モデルを学習する [1]。NMT でも、各単語はすべての語彙からのマルチクラス分類によって選択されるため、上記の知識蒸留手法が適用できる（単語レベル知識蒸留と呼ぶ）。

しかし、同様な効果を持ち、さらにシンプルな方法として、教師の翻訳結果で疑似対訳を作成し、それを基に生徒モデルを訓練する方法も提案されている（系列レベル知識蒸留と呼ぶ）[2]。系列レベル知識蒸留は、データの加工のみで知識蒸留を実現し、モデルや損失関数は既存の方式を使えるという利点がある。機械翻訳で知識蒸留と言った場合、系列レベル知識蒸留を指すことが多い。



(a) 系列レベル知識蒸留　(b) 逆翻訳

図 1　知識蒸留と逆翻訳の手順

なお、知識蒸留の目的は、教師モデルの知識をできる限り正確に生徒モデルに移すことであるので、一般的には生徒モデルが教師モデルの精度を超えることはない[1)]。

2.2 逆翻訳との関係

基になる機械翻訳器を使って、疑似対訳を作成するという点で、系列レベル知識蒸留（以下、単に知識蒸留と呼ぶ）と、逆翻訳法 [6] は似ている（図 1）。最大の違いは、逆翻訳が目的言語を原言語に翻訳することで疑似対訳を生成するのに対して、知識蒸留は原言語を目的言語に翻訳した疑似対訳を使う点である。疑似対訳を作成するデータについては、逆翻訳は対訳コーパスとは異なる単言語コーパスを用いることを前提としているのに対して、知識蒸留は、使用するコーパスを規定していない。そのため、教

1） 本稿の実験結果のように、元の訓練データに不備がある場合、それが修正されることで精度が向上する場合はある。

表 1　知識蒸留と逆翻訳の対比

	知識蒸留	逆翻訳
疑似対訳の翻訳方向	原言語 → 目的言語	目的言語 → 原言語
使用コーパス	教師モデルと同じコーパス（対訳）を使う場合が多い。	教師とは別の単言語コーパスを使う場合が多い。
対訳コーパスを使う場合の人手翻訳方向	原言語が原文 (original) で目的言語が人手翻訳文 (translationese) の場合に効果大。	目的言語が原文で、原言語が人手翻訳文の場合に効果大 [3]。
教師モデルの品質	教師モデルの品質が、直接生徒モデルの品質に影響するため、できるだけ高品質なモデルを用いる。	多様性が重要であるため、疑似対訳作成時にサンプリング、ノイズ混入、複数逆翻訳で多様性を増強させる [4, 5]。

師モデルの訓練に使用した対訳コーパスの原言語部分を使う場合も多い。

知識蒸留と逆翻訳の翻訳方向が逆であるという特徴は、人手作成した対訳コーパスの翻訳方向によって効果が変わってしまうという特徴にも繋がる。

文献 [3] では、人手翻訳の方向と逆翻訳の関係を調べ、逆翻訳は目的言語が original（以下「原文」と訳す）で、原言語が translationese（以下人手翻訳文と訳す）の場合に BLEU が向上することを確認した[2)]。もともと逆翻訳はデコーダーを強化するために開発された手法なので、目的言語に自然な文を与えると翻訳品質は向上する。逆に原言語はさまざまな入力に対応する必要があるので、多様性が求められる。そのため、疑似対訳作成時にサンプリング、ノイズ混入、複数逆翻訳で多様性を増強させると、翻訳品質が向上する [4, 5]。

一方、知識蒸留では、原言語が原文で、目的言語が人手翻訳文の方が効果が高まる。これは、目的言語の翻訳文が均質（多様性が少ない）になるため、デコーダーの学習効果が高まるためと考えられる。また、知識蒸留は教師モデルの出力を模倣させるため、教師モデルの品質が直接生徒モデルの品質に影響する。そのため教師モデルにはできるだけ高品質なものを用いるのが望ましい。知識蒸留と逆翻訳の対比を表 1 にまとめる。

2.3　高速化方法

本稿では、モデルの縮小（パラメーター削減）による高速化と、探索空間の縮小による高速化について検討する。

モデルを縮小するためには、(a) 深層モデルのレイヤー数を削減する方法、(b) 分散表現の次元数を削減する方法がある。

また、ニューラル機械翻訳に特有のパラメーター削減方法としては、(c) 語彙サイズを縮小する方法がある。語彙サイズを縮小すると、デコーダーの出力層の SoftMax 操作の計算量が削減され、高速化される。近年、NMT ではサブワードを用いることが一般的になっているが、これは語彙数を指定して分割するため、語彙数を制御することが可能である。本稿では、SentencePiece（unigram モデル）[7] によるサブワード化で語彙を縮小する。

探索空間の削減は、(d) 翻訳時のビーム幅を縮小することで高速化を行う。

3　実験

3.1　実験条件

言語対・コーパス　本稿では、日本語と英語の科学技術文献対訳コーパスである ASPEC-JE[8] を使用する。このコーパスは、約 300 万文の訓練セット、1,812 文のテストセットなどで構成されている。ASPEC は、日本語の文献を英語に翻訳したものなので、日本語が原文で、英語が人手翻訳文である。

翻訳システム・環境　本稿では、fairseq 翻訳器 [9] を使用する。これは、PyTorch 上に作られた翻訳器で、Transformer [10] を含んでいる。テストに使用したハードウェアは、Intel Xeon Gold 6150 CPU (2.7GHz)、NVIDIA V100 GPU (32GB) 1 個で、OS は CentOS 7.9 である。

教師モデル・知識蒸留　教師モデルは、ASPEC の訓練セットをすべて使用して、ランダムシードを変えた Transformer big モデル（6 層、モデル次元数 1,024, FFN 次元数 4,096）を 4 個作成した。なお、教師モデルの訓練時には、SentencePiece を用いて、日本語、英語それぞれ約 1.6 万のサブワードに分割した。

知識蒸留には、ASPEC の訓練セットすべての原言語側を使用し、上記 4 モデルのアンサンブルで翻訳し、疑似対訳を作成した。

2)　彼らの実験では、原言語が原文で、目的言語が人手翻訳文の場合は BLEU に対する効果は少なかった。しかし、主観評価では、翻訳方向に関わらず効果があった。

表 2　翻訳方向による知識蒸留の効果

モデル	日英 BLEU 直接	蒸留	差	英日 BLEU 直接	蒸留	差
Base	27.78	29.81	+2.03	41.26	41.71	+0.45
Big	28.58	30.00	+1.42	41.47	41.77	+0.30

生徒モデル　今回使用する生徒モデルは、Transformer base モデル（6 層、モデル次元数 512、FFN 次元数 2,048）を基本にし、設定を以下のとおり可変にする。

(a)エンコーダー、デコーダーのレイヤー数をそれぞれ 1, 2, 4, 6 層にした場合。

(b)モデル次元数を 128, 256, 512, 1024 にした場合。なお、FFN 次元数はモデル次元数の 4 倍とし、ヘッド数は 64 次元固定とする。つまり、モデル次元数を変えるとヘッド数が変わることになる。

(c)語彙数を 4 千、8 千、1.6 万、3.2 万、6.4 万にした場合。2.3 節で述べたように、語彙数は SentencePiece によるサブワード化で制御する。

(d)生徒モデルとしては Transformer base モデルを用いるが、テスト時のビーム幅を 1, 2, 3, 5, 7 ,10, 20 と変えた場合。

なお本稿では、知識蒸留による疑似対訳を使用した生徒モデルを知識蒸留モデル、ASPEC 訓練セットを直接使って作成したモデルを直接モデルと呼称する。

評価指標　本稿では、翻訳品質の評価は case sensitive BLEU [11] で行う。これは Workshop on Asian Translation [12] の評価方法と同じである。

速度の評価は、1 秒あたりの翻訳トークン数で行う。これは、テストセットのトークン数を翻訳時間（初期化を含まない）で割ったものである。

3.2　実験結果

3.2.1　日英翻訳 vs. 英日翻訳

まず、翻訳方向による知識蒸留の効果を確認するため、日英翻訳 (original → translationese) と英日翻訳 (translationese → original) の翻訳品質を確認する。

結果を表 2 に示す。知識蒸留の効果は、日英では Base モデルで+2.03 あるのに対して、英日では+0.45 しかなく、知識蒸留は翻訳方向によって、効果に大きな差があることがわかる。

なお、ASPEC データは、日英の対訳記事から自動で文アライメントを作成しているため、対訳文として（そもそも単語が対応しない）不適切なデータも含まれている。英日翻訳で BLEU が向上した理由は、不適切な対訳文が新たに生成されることで対訳文の品質が向上したためと考えられる。

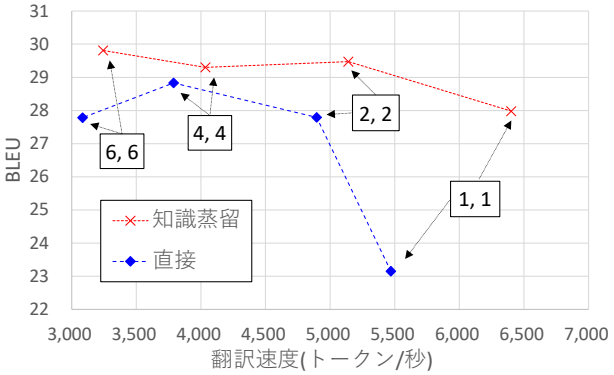


図 2　知識蒸留と直接モデルの比較 ((a) レイヤー削減)

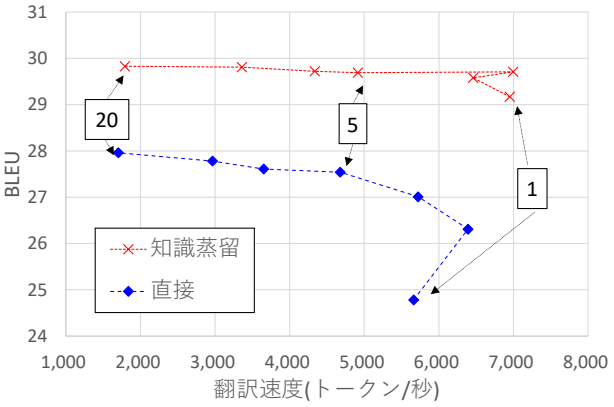


図 3　知識蒸留と直接モデルの比較 ((d) ビーム幅)

以下、知識蒸留の効果が大きい日英翻訳に限定して、実験を行う。

3.2.2　知識蒸留モデル vs. 直接モデル

次に、知識蒸留の高速化への効果を確認するため、(a) レイヤー削減、(d) ビーム幅変更について、直接モデルと知識蒸留モデルの翻訳品質と速度を比較する。

図 2 は (a) レイヤー削減について、直接モデルと知識蒸留モデルを比較したグラフである。横軸が翻訳速度、縦軸が BLEU スコアを表している。ボックス内の数字はそれぞれエンコーダー、デコーダーのレイヤー数である。どちらのモデルも、レイヤー数を削減するに伴い、翻訳速度が向上し、BLEU スコアは低下する傾向がある。しかし、BLEU スコアに着目すると、6 レイヤーから 1 レイヤーに削減した場合、直接モデルでは 4.63 低下したのに対して、知

表 3　エンコーダー・デコーダーのレイヤー数別の翻訳速度と翻訳品質

エンコーダー＼デコーダー	1		2		4		6	
	BLEU	速度	BLEU	速度	BLEU	速度	BLEU	速度
1	27.98	6402	29.16	4907	29.45	3790	29.21	2868
2	28.98	6382	29.47	5141	28.99	3327	29.38	3273
4	29.08	6368	29.63	5490	29.30	4034	30.11	3344
6	29.28	6517	29.58	5449	29.66	4137	29.81	3243

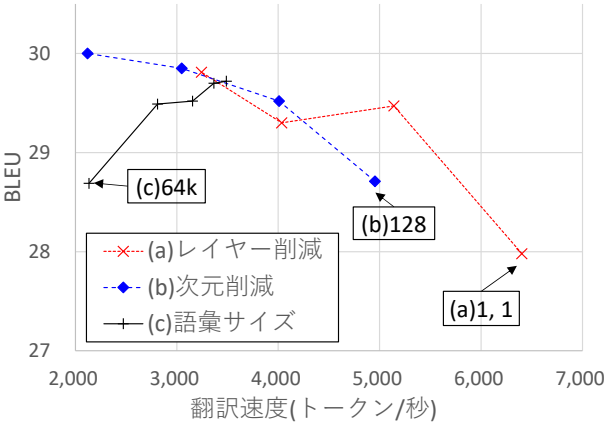


図 4　モデル縮小方式ごとの翻訳速度と翻訳品質

識蒸留モデルでは 1.83 の低下で収まった。

一方、図 3 は、(d) ビーム幅について同様に比較したグラフである。ボックス内の数字はビーム幅を表す。このグラフでは、ビーム幅を狭くするのに伴い翻訳速度が向上するが、BLEU スコアについては、ビーム幅を 20 から 1 に変更した場合、直接モデルでは 3.18 低下したのに対して、知識蒸留モデルでは 0.66 の低下にとどまった。

知識蒸留モデルは、目的言語の多様性を低減する方式であるため、モデルパラメーターを少なくしたり、探索空間を狭くしても、翻訳品質の劣化が少ない方式であると言える。

3.2.3　高速化に適したモデル縮小法

本節では、(a) レイヤー削減、(b) 次元数の削減、(c) 語彙サイズについて、知識蒸留と組み合わせた場合の翻訳品質と速度を検証する。

知識蒸留モデルにおける翻訳速度と翻訳品質の関係を図 4 に示す。各データポイントの詳細は省略しているが、モデル縮小方式ごとの傾向は見てとれる。これによると、まず、高速化を目標とした場合、レイヤー削減による高速化が最も効果が大きく、次元削減による高速化の効果は限定的である。

また、語彙サイズに関しては、大きい場合に翻訳品質が低下する傾向が得られたが、これを小さくしても速度は大きく向上しない。

レイヤー削減が高速化に高い効果がある傾向が得られたため、次に、これを詳細に分析する。表 3 は、エンコーダーとデコーダーのレイヤー数を独立に変更したときの翻訳品質と速度の測定結果である。

まず、BLEU スコアに着目すると、エンコーダー・デコーダーともに 6 層の条件から、エンコーダーのみ 1 層に減らした場合、デコーダーのみ 1 層に減らした場合、それぞれ 29.21、29.28 であるので、どちらを減らしても翻訳品質への影響は少ない。

一方、翻訳速度に着目すると、エンコーダーのレイヤー数を変更しても、速度への影響は少ない。逆にデコーダーのレイヤー数を削減すると、翻訳速度を速くすることができる。これは、Transformer のエンコーダーは 1 ステップですべての入力トークンを処理するのに対して、デコーダーは自己回帰型 (autoregressive) 方式で、1 トークンずつ生成するためである。まとめると、レイヤー数を削減する場合、デコーダーを優先する方が高速化効果が高い。

4　まとめ

知識蒸留は、目的言語の多様性を削減するため、翻訳品質をあまり落とさずに高速化ができる。本稿で確認した知見は以下のとおりである。

- 知識蒸留は、翻訳方向が原文 → 人手翻訳文（ASPEC の場合、日英翻訳）のときに効果が大きい。
- 知識蒸留を行うことで、翻訳品質の低下を抑えて高速化することができる。
- モデルの縮小方式には、レイヤー数削減が高速化に効果的である。
- レイヤー数を削減する場合、エンコーダーよりデコーダーを削減した方が効率的に高速化が可能である。

この検討結果を踏まえて、NMT を構築してゆく。

参考文献

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.
[2] Yoon Kim and Alexander M. Rush. Sequence-level knowl-

edge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016.

[3] Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. On the evaluation of machine translation SystemsTrained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2836–2846, Online, July 2020.

[4] Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 55–63, 2018.

[5] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, 2018.

[6] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, 2016.

[7] 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, Melbourne, Australia, July 2018.

[8] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 2204–2208, Portorož, Slovenia, may 2016.

[9] 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 NAACL-HLT 2019: Demonstrations*, 2019.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, Vol. abs/1706.03762, , 2017.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, Philadelphia, Pennsylvania, USA, 2002.

[12] Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, pp. 1–44, Suzhou, China, December 2020.