ニューラル機械翻訳のためのノイズ寛容なアンカー学習

根石 将人
東京大学大学院 情報理工学系研究科
neishi@tkl.iis.u-tokyo.ac.jp

吉永 直樹
東京大学 生産技術研究所
ynaga@iis.u-tokyo.ac.jp

概要

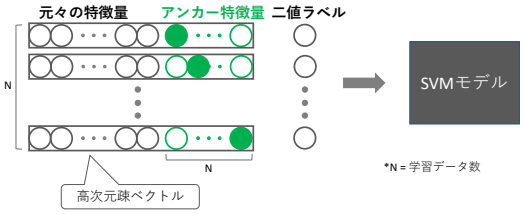
機械翻訳モデルの学習に用いられる大規模な学習データは収集方法ゆえにノイズデータを含み、それらはモデルの性能低下を引き起こす。本研究は、SVM モデルにおいて提案されたノイズデータがモデルに及ぼす悪影響を低減するアンカー学習手法に注目し、これをニューラル機械翻訳 (NMT) に導入することを試みる。実装上問題となる計算コストを抑えるアンカー表現手法を、ランダムグルーピング手法と組み合わせ手法の 2 通り提案し、ASPEC 英日翻訳と WMT2017 英独翻訳の 2 つの翻訳タスクでその効果を確認した。

1 はじめに

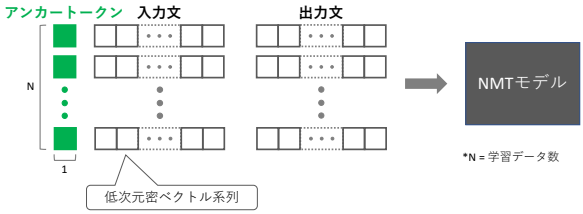
高品質な NMT モデルの学習には大規模な対訳文対データが必要であるが、そのようなデータは Web クローリングなどで自動収集されているために対訳関係にないノイズデータが含まれ、これが翻訳モデルの品質低下を引き起こすことが知られている [1]。

この問題に対処するため、Conference on Machine Translation (WMT) では 2018〜2020 年にかけて対訳コーパスフィルタリングに関する Shared Task が開催された [2, 3, 4]。これらの Shared Task を通して提案されたフィルタリング手法を用いて、Herold ら [5] は独語から英語の翻訳タスクにおいて合成混入させたノイズデータの内多くの種類について 90%以上の除外に成功したと報告している。しかしながら、ノイズの種類によっては除外精度が 70%程度に過ぎず、またクメール語から英語のタスクにおいては全体的に精度が低下した。現状のフィルタリング手法ではノイズデータは除外しきれず、また認識されている種類以外のノイズデータの存在も考えられる。

そこで本研究では、Goldberg ら [6] がサポートベクタマシン (SVM) [7] を用いた分類タスクのために提案したアンカー学習に注目した。SVM は二値分類を行う識別モデルで、学習データとして特徴量ベクトルと二値分類ラベルを使用する。アンカー学習では、この特徴量ベクトルに各学習データ固有の新たな特徴量 (アンカー特徴量) を追加し拡張する (図 1a)。この新たに追加されたデータ固有の特徴量によって、各データがその他のデータに対して線形分離可能になる。これにより SVM モデルは、元々の特徴量から出力の推論が難しい場合においても、アンカー特徴量から正解の出力を推論することを丸覚えする選択肢が与えられる。こうしてアンカー学習は、誤った入出力関係をもつノイズデータからの学習におけるモデルへの悪影響を低減し、モデルの性能を向上させることが期待できる。



(a) SVM モデルにおけるアンカー学習 [6]



(b) NMT モデルにおけるアンカー学習 (提案手法)

図 1: 2 つのモデルにおけるアンカー学習

なお、正確にはアンカー学習は入力から出力の推論が難しい場合 (hard-to-learn cases) に対応しており、これは翻訳タスクでの例としては低頻度な慣用句などの厳密にはノイズデータではない場合も含む。本研究では便宜的にそのような場合も含めノイズデータと呼称し、一方で入力から出力の推論が易しいデータを綺麗なデータと呼称する。

本研究では、系列変換モデルを用いた機械翻訳タスクを対象としたアンカー学習を提案する。機械学習モデルおよび入力データ形式の違いから、個々の学習データを弁別化するアンカーとして、アンカー特徴量ではなくアンカートークンを導入する。また、ナイーブにはアンカーは学習データと同数用意する必要があるが、大規模な学習データを用いる場合にアンカートークンの数が爆発し膨大な計算資源を必要とする問題がある。この解決のために本研究では、ランダムグルーピング手法と組み合わせ手法の 2 種類のアンカートークン手法を提案する。

実験は ASPEC 英日翻訳と WMT2017 英独翻訳の 2 つの翻訳タスクでノイズデータを混合したデータセットで行い、組み合わせ手法によるアンカー学習が、計算コストの増加を抑えつつ翻訳の BLEU スコアを向上させることを確認した。

2 系列変換モデルのためのアンカー学習

本研究では、SVM 分類タスクで提案されたアンカー学習を、系列変換モデルを用いた機械翻訳タスクに導入するのだが、機械学習モデルや学習データの形式およびデータ数の違いにより、そのままでは適用出来ない。SVM モデルの入力データは高次元疎ベクトルであり、一方で系列変換モデルの入力は低次元密ベクトルの系列である。また、SVM による分類タスクに比べて機械翻訳のような生成タスクは、一般に学習データ数が大きい。

まず機械学習モデル及びデータの形式の違いに対応するため、アンカー特徴量の代わりにアンカートークン (もしくはアンカーベクトル) を導入する (図 1b)。アンカートークンは、モデルの通常入力である単語やサブワードなどと同様に扱い、通常入力に連結することでモデルへ入力する。一方、アンカーは基本的には学習データと同数必要なため、大規模な学習データに対してはアンカーの数が大きくなり、それに対応してアンカートークンをベクトルへ変換する埋め込み層のパラメータ数が増大し、学習時の計算コストが莫大になる問題がある。この学習データサイズの問題に対して、用意するアンカートークン数を削減する 2 つの手法を提案する。

2.1 アンカートークン数削減のためのアンカー表現手法

ランダムグルーピング手法　アンカートークン数削減の一つ目の手法として、一つのアンカートークンを複数の学習データ間で共有するランダムグルーピング手法 (RGA: Random Grouping Anchor) を提案する。この手法では用意するアンカートークンの数 N_g をハイパーパラメータとして設定し、各学習データに対してランダムにアンカートークンを割り当てる。

アンカーは本来学習データと同数求められるが、学習データはノイズデータのみならず綺麗なデータも含み、この綺麗なデータにはアンカーは不要である。またノイズデータと綺麗なデータが同一のアンカーを共有している場合、アンカーがノイズデータの推論に役立つよう学習されることで、逆に綺麗なデータについてはアンカーを無視するような学習が促進される効果も期待できる。しかしながら複数のノイズデータが同一のアンカーを共有する可能性もあり、この場合は期待されるノイズ低減効果は得られず、用意するアンカートークンの数については学習データ中のノイズデータの数や割合を念頭に適切に調整する必要がある。

組み合わせ手法　ランダムグルーピング手法はアンカートークンの数を大幅に削減するものの、アンカートークン数は学習データに対して比例の関係に過ぎない。そこで学習データのさらなる増加に対応出来るように、単一のアンカートークンではなく、複数のアンカートークンの系列をアンカーとして使用する組み合わせ手法 (CA: Combination Anchor) を提案する。組み合わせ手法では、各学習データに固有のアンカーを用意するために、限られた数のアンカートークンを用意し、それらを組み合わせたアンカートークン系列をアンカーとして利用する。例えば 32 個のアンカートークンを用意し、系列長を 3 とした場合、$32^3 = 32768$ 個までの学習データに対して固有アンカーを割り振ることが出来る。本研究の実装ではハイパーパラメータとしてアンカートークン数 N_c のみを設定し、系列長は学習データ数に足りる最小値を採用した。

組み合わせ手法では系列を構成するアンカートークンはそれぞれ複数回使用されるため、アンカートークン系列同士も類似した表現になる可能性がある。この問題の解決、さらにランダムグルーピング手法と条件を揃える為に、組み合わせ手法には回帰型ニューラルネットワーク (RNN) を導入し、アンカートークン系列を単一のアンカー表現に変換する。結果として NMT モデルのパラメータ数は若干

増加するが、学習データと同数のアンカートークンを用意する場合やランダムグルーピング手法と比較して、増加量は大幅に抑えられる。

2.2 アンカートークンの入力方法

NMT モデルへのアンカートークンの入力箇所として、エンコーダ側の入力の先頭および最後尾、またデコーダ側の入力の先頭の 3 箇所が候補として考えられる。本研究では、テキスト以外の情報を系列に追加する先行研究 [8, 9, 10, 11] を踏まえ、先頭を採用する。またエンコーダ側へのアンカーの入力では、アンカー情報がエンコーダ内部で入力文との相互処理を経て文脈化され、アンカーがより有用な情報を所持することが期待できる。デコーダ側先頭への入力はこれが期待できないため、本研究ではエンコーダ側の入力の先頭を入力箇所として採用する。

アンカーによる推論を補助する情報の追加は、所謂 prompt 手法 [12] と類似しており、特に追加する情報自体を学習するという点では soft prompt [13] と同じである。しかしながらメインのモデルの学習と同時に追加情報であるアンカーを学習する点、またタスク毎ではなく学習データの事例毎に追加情報を用意する点に違いがある。

先行研究 [6] を参考に、アンカートークンは学習時のみに使用し、推論時は通常通りアンカートークンは存在しないまま処理を行う。本研究では NMT モデルとして Transformer[14] ベースのモデルを利用するが、オリジナルのモデルで使用する絶対的な位置情報では、学習時には常にアンカートークンが特定の位置にあるのに対して、推論時にその特定の位置にアンカートークン以外のトークンが現れるという不整合が起きる。これを避けるために本研究では相対位置に基づく Transformer [15] を採用する。

3 実験設定

提案手法が、NMT モデルの学習においてノイズデータの悪影響を低減する効果を確かめるため、2 つの翻訳タスクにおいて実験を行う。学習データとしては、綺麗と見做されているデータセットとノイズデータが多いと見做されているデータセットの両方を用意する。基本の実験として両方のデータセットを混合したデータセットを用いるが、綺麗と見做されているデータセットにもノイズデータが含まれていることを考慮し、これを単一で用いた場合での実験も行う。

3.1 モデル

NMT モデルとして、PyTorch[1] (ver. 1.12.0) で実装した相対位置を利用する Transformer[15] を用いる。提案手法の実装のために、アンカートークンおよびアンカートークン用の単語埋め込み層、また組み合わせ手法における RNN として単一の GRU 層をエンコーダに追加実装した。モデルのハイパーパラメータについては概ね Vaswani らの Base モデル [14] に従い、また相対位置ベクトルの最大距離についてのパラメータ k は 16 とした。

提案手法に関するハイパーパラメータとして、ランダムグルーピング手法については、用意するアンカートークンの数がノイズデータとされるデータの数のおよそ 1, 10, 100%となる場合を実験する。組み合わせ手法のアンカートークン数については、アンカートークン系列長が 3〜4 程度となる $N_c = 128, 256, 512$ の場合を実験する。

3.2 データセット

翻訳タスクとして ASPEC[16] による科学技術論文ドメインの英日翻訳と、WMT2017 [17] によるニュースドメインの英独翻訳を用いる。

ASPEC データセットについては、英語のデータは Moses toolkit[2] (ver. 2.2.1) を用いてトークン化及び Truecasing を行い、日本語のデータは KyTea[3] (ver. 0.4.2) による単語分割を行った。以上の処理の後、SentencePiece[18] により両言語合わせて語彙数を 16000 として unigram モデルにてサブワード化した。ASPEC の学習データは対訳文の類似度が高い順に並べられている[4]ため、前半 150 万文を綺麗なデータセットとして、また後半 150 万文をノイズの多いデータセットとして採用した。

WMT2017 データセットについては、公式に配布されている前処理済みデータ[5]を使用し、newstest2015 を開発データ、newstest2016 をテストデータとした。ノイズの多いデータセットとしては ParaCrawl[19] を採用し、Moses toolkit を用いて前処理済みデー

1) https://pytorch.org/
2) http://www.statmt.org/moses/
3) http://www.phontron.com/kytea/
4) データに付属する README に、「訓練データは、類似度の高い対訳文から順に並べて、上位 100 万文を train-1、次の上位 100 万文を train-2 とし、残りを train-3 とした。それゆえ、train-2 や train-3 に含まれるデータは、train-1 と比べると対訳文としての質が低い。」とある。
5) https://data.statmt.org/wmt17/translation-task/preprocessed/

表 1: ASPEC 英日翻訳における BLEU スコアおよびノイズ混入データでのモデルのパラメータ数

		綺麗のみ	ノイズ混入	パラメータ数
通常学習		42.08	41.03	69M
RGA	N_g = 10k	42.47	**42.15**	74M
	100k	42.29	41.50	120M
	1M	42.45	41.61	485M
CA	N_c = 128	42.17	41.74	70M
	256	42.50	41.59	70M
	512	**42.53**	41.42	71M

表 2: WMT2017 英独翻訳における BLEU スコアおよびノイズ混入データでのモデルのパラメータ数

		綺麗のみ	ノイズ混入	パラメータ数
通常学習		29.81	31.07	106M
RGA	N_g = 30k	30.26	**31.50**	121M
	300k	**30.34**	31.16	259M
	3M	30.04	31.22	1.53B
CA	N_c = 128	29.99	**31.50**	107M
	256	30.16	31.37	107M
	512	30.21	30.98	108M

タセットと同様の前処理を行った。以上の処理の後に、語彙数を 40000 として ASPEC と同様のサブワード化を行った。

両データセット共に、学習時には最大文長を 100 とし、それを超えるデータは排除した。

3.3 ノイズ混入データ

現実的なノイズデータが混ざった学習データの条件の設定するために、綺麗なデータセットとノイズの多いデータセットの混合データセットを用意する。Khayrallah ら [1] の実験結果を踏まえ、ノイズデータによるモデルの性能低下を確認できるよう綺麗なデータとノイズデータの割合は 100:50 とした。ASPEC 英日翻訳では先頭 150 万文と後尾 75 万文を混ぜ、また WMT2017 英独翻訳では配布データセットの全 590 万文と ParaCrawl の先頭 300 万文を混ぜて、ノイズが混入した学習データを作成した。

3.4 評価

前処理としてサブワード化を行ったため、評価時には単語単位での評価のための処理を行う。ASPEC 英日翻訳では出力を全て連結した後に KyTea による単語分割を再度行い、WMT2017 英独翻訳では出力に対して Moses toolkit による Detokenization を行った。評価には sacreBLEU[20] を用いた BLEU スコア [21] を使用した。学習は 30 万ステップ行い、1 万ステップ毎に開発データでのスコアを算出し、それが最も高い時点のモデルを最終的なテストデータでのスコア算出に用いた。BLEU スコアの算出のための NMT モデルの出力は全て貪欲法にて行った。

4 実験結果

ASPEC 英日翻訳における結果を表 1 に、WMT2017 英独翻訳における結果を表 2 に示す。WMT2017 ノイズ混入データにおける組み合わせ手法 (N_c = 512) のみ通常学習を下回るスコアであった。それ以外では、どちらの言語対においても、またデータセットのノイズ含有度合いに関わらず、通常の学習に比べてアンカー学習がより高いスコアを出した。アンカー学習の効果が概ね確認され、さらに綺麗と見做されているデータセットでもノイズデータの混入によるモデルの精度低下が認められた。

今回の結果では、2 つのアンカー数を抑えた提案手法に BLEU スコアでの優劣差は認められず、それぞれのハイパーパラメータについても明確な傾向は掴めなかった。これらについては学習の複数回試行を含め、より厳密な分析が必要である。

提案手法によるモデルのパラメータ数の変化について、表 1、2 右端列にノイズ混入データセットにおける実験での NMT モデルのパラメータ数を示す。通常モデルにおける ASPEC 英日翻訳と WMT2017 英独翻訳のパラメータ数の違いは語彙数の違いによるものである。ランダムグルーピング手法ではアンカートークン数が大きい場合にパラメータ数が大幅に増加している一方で、組み合わせ手法は GRU 層の追加によるパラメータ数の増加がありつつも、アンカートークン数は大幅に少ないため全体的には数%程度の増加に留まっている。結果としてランダムグルーピング手法は学習データが大きい場合には実用的ではなく、組み合わせ手法のみが実用的な範囲に収まっている。

5 おわりに

本研究は NMT の学習において、ノイズデータの悪影響を低減するアンカー学習の手法を導入した。計算コストの面からアンカートークン数を抑える手法として、ランダムグルーピング手法と組み合わせ手法の 2 つを提案し、ASPEC 英日翻訳と WMT2017 英独翻訳の 2 つの翻訳タスクにてその効果を確認した。2 つの提案手法による BLEU スコアの向上幅は同程度であり、パラメタ数増加の観点から、組み合わせ手法の方が優れた手法であると言える。

謝辞

本研究は JSPS 科研費 JP21H03494 の助成を受けたものです。

参考文献

[1] H. Khayrallah and P. Koehn. On the impact of various types of noise on neural machine translation. In **Proceedings of the 2nd Workshop on Neural Machine Translation and Generation**, pp. 74–83, 2018.

[2] P. Koehn, H. Khayrallah, K. Heafield, and M. L. Forcada. Findings of the WMT 2018 shared task on parallel corpus filtering. In **Proceedings of the Third Conference on Machine Translation: Shared Task Papers**, pp. 726–739, 2018.

[3] P. Koehn, F. Guzmán, V. Chaudhary, and J. Pino. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In **Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)**, pp. 54–72, 2019.

[4] P. Koehn, V. Chaudhary, A. El-Kishky, N. Goyal, P. Chen, and F. Guzmán. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In **Proceedings of the Fifth Conference on Machine Translation**, pp. 726–742, 2020.

[5] C. Herold, J. Rosendahl, J. Vanvinckenroye, and H. Ney. Detecting various types of noise for neural machine translation. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 2542–2551, 2022.

[6] Y. Goldberg and M. Elhadad. SVM model tampering and anchored learning: A case study in Hebrew NP chunking. In **Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics**, pp. 224–231, 2007.

[7] V. N. Vapnik. **The nature of statistical learning theory**. 1995.

[8] D. Britz, Q. Le, and R. Pryzant. Effective domain mixing for neural machine translation. In **Proceedings of the Second Conference on Machine Translation**, pp. 118–126, 2017.

[9] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. **Transactions of the Association for Computational Linguistics**, Vol. 5, pp. 339–351, 2017.

[10] S. Sato, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa. Modeling situations in neural chat bots. In **Proceedings of ACL 2017, Student Research Workshop**, pp. 120–127, 2017.

[11] Y. Wang, C. Hoang, and M. Federico. Towards modeling the style of translators in neural machine translation. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1193–1199, 2021.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901, 2020.

[13] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3045–3059, 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems (NIPS) 30**, pp. 5998–6008, 2017.

[15] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 464–468, 2018.

[16] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In **Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)**, pp. 2204–2208, 2016.

[17] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi. Findings of the 2017 conference on machine translation (WMT17). In **Proceedings of the Second Conference on Machine Translation**, pp. 169–214, 2017.

[18] T. 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.

[19] Marta Bañón, P. Chen, B. Haddow, K. Heafield, H. Hoang, M. Esplà-Gomis, M. L. Forcada, A. Kamran, F. Kirefu, P. Koehn, S. Ortiz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarrías, M. Strelec, B. Thompson, W. Waites, D. Wiggins, and J. Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4555–4567, 2020.

[20] M. Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, 2018.

[21] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, 2002.

A　NMT モデルの学習におけるハイパーパラメータ

NMT モデル	Transformer with self-attention with relative position [15]
相対位置ベクトルの最大距離 k	16
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 0.0001$)
学習率スケジュール	Vaswani ら [14] による warm up 手法
Warm up ステップ数	4000
Dropout 率	0.2
Gradient Clipping	3.0
ミニバッチサイズ	128
最大文長	100
学習ステップ数	300,000