モデルに基づくマスキングを行う非自己回帰的ニューラル機械翻訳

安井 豪 † 鶴岡 慶雅 † 永田 昌明 ‡
† 東京大学 大学院情報理工学系研究科
‡NTT コミュニケーション科学基礎研究所
†{gyasui,tsuruoka}@logos.t.u-tokyo.ac.jp
‡masaaki.nagata.et@hco.ntt.co.jp

1 はじめに

ニューラルネットワーク技術の発展に伴い、機械翻訳においても統計的手法に代わり、ニューラル機械翻訳を用いることが一般的になってきた [1, 8]。ニューラル機械翻訳の中でも Transformer [8] を用いた手法は高い精度を誇る一方で、従来の再帰的なニューラルネットワークを使った手法と比べ、出力の計算に時間がかかることが知られている。

Transformer は、学習時には並列に複数のトークンを出力できるが、推論時には LSTM[4] 等と同様に自己回帰的にトークンを計算する必要がある。LSTM 等と比べ一つのトークンに対する計算時間が長いことが推論時の計算時間の長さの原因となっている。また、強化学習のような学習時にも推論時の挙動を再現する必要がある手法は、学習に時間がかかりすぎるため、Transformer ではあまり用いられない。

計算時間の問題は、強化学習の是非だけではなく、実用面にも大きな影響を与えている。推論時の計算時間を短縮するため、Knowledge Distilaltion [10] を用いてモデルのパラメータを減らすなど、様々な高速化の手法が提案されている。中でも特に注目を集めている高速化の手法が、復号の仕組みを変え、Decoder を非自己回帰的にすることである。

非自己回帰的機械翻訳モデルでは、ひとつずつトークンを生成するのではなく、文中のすべてのトークンを並列に生成する。この場合一文あたりの合計計算回数が少なく、復号の高速化が期待できる。一度の並列計算で正確に文を生成することは難しいため、Mask-Predict [2] や Levenshtein Transformer [3] といった精度の高いとされている手法では何度か復号の計算を繰り返し生成文を改善する。

これら非自己回帰的手法の特徴のひとつは、学習時のアルゴリズムと推論時のアルゴリズムが異なる点である。学習時の挙動を推論時の挙動に近づけることは性能向上に繋がるとされているため、このギャップは欠点になりうる [9]。また、強化学習のような手法を適用しづらく、復号の高速化を性能向上にも利用することが難しい。本研究では、Mask-Predict をベースとし、そのマスキング機構に変更を加えることで、非自己回帰的機械翻訳においても学習時に推論時の挙動を再現する手法を提案する。

2 関連研究

2.1 非自己回帰的翻訳モデル

自己回帰的な文生成を行うモデルでは文を先頭から順番に 1 トークンずつ生成していく。一方で非自己回帰的文生成モデルでは、トークン生成に順番はなく、すべてのトークンが並列に生成される。Transformer は再帰的な構造を持たないが代わりに過去のトークンすべてを入力として計算することで、次のトークンを生成する。非自己回帰的 Transformer では、すべてのトークンを並列に計算するため、学習時に未来のトークンを見ないためにかける Causal Mask が存在しない。また、自己回帰的機械翻訳では、文末トークンが出た時点で出力を止めるが、非自己回帰的機械翻訳ではトークン生成の前にあらかじめ出力の長さ L を決めておく必要がある。これらを踏まえると非自己回帰的機械翻訳での計算は式 (1) のようになる。

$$p_{NA}(Y \mid X) = p_L(L \mid X) \prod_{l=1}^{L} p_Y(y_l \mid L, X) \quad (1)$$

ここで、$X = [x_1, x_2, ..., x_t, ..., x_T]$ は翻訳元言語の入力系列、$Y = [y_1, y_2, ..., y_l, ..., y_L]$ は翻訳先言語の出力

系列、$p_L(\cdot)$ は出力長の予測モデル、$p_Y(\cdot)$ は出力トークンの予測モデルを表す。

2.2 Conditional Masked Language Model

Conditional Masked Langage Model (CMLM) [2] は一部マスクトークンによって隠された翻訳先言語の入力系列を元に、隠された位置に対応するトークンを予測する非自己回帰的文生成モデルである。式 (1) と比較すると、CMLM ではトークン予測モデルの入力に一部マスクのかかった翻訳先言語のトークン系列が加わる。つまり、翻訳元言語の入力系列を前提条件として、翻訳先言語でトークンの穴埋め問題を解くことになる。学習時にはランダムにマスクで隠した教師系列を入力とし、損失の計算には一般的な翻訳モデルと同様に、教師系列との Cross-entropy 損失を用いる。このときマスクがかかっていた位置での損失のみを計算する。

2.3 Mask-Predict

Mask-Predict [2] は CMLM を使って文を生成する際のアルゴリズムである。Mask-Predict による復号は次のように行う。

1. p_L により出力長を予測し、すべての位置をマスクで隠した入力系列を用意する。
2. CMLM でマスクのかかった位置のトークンを予測する。
3. CMLM の出力のうち予測尤度が低いものに対してマスクをかけ次の復号ステップでの入力とする。
4. 予め決めておいた復号ステップ回数に達するまで 2. と 3. を繰り返す。

なお、3. においてマスクで隠すトークンの数は最後の復号ステップが終わる際にゼロになるよう線形に減らしていく。

3 提案手法

前述のとおり、Mask-Predict では 2 ステップ目以降の各復号ステップにおいて、前ステップでの出力系列のうち尤度が低いトークンを隠して入力系列とする。したがって、学習時にはランダムに、推論時には尤度基準にマスクがかかることになり、CMLM の入力の性質に学習時と推論時の間でギャップが生じる。例えば、推論時に発生しがちな同一トークンの繰り返しは学習時の入力データには存在しない。本研究では、学習時にも推論時の復号過程を再現して学習させるため、マスキング機構のモデルを提案する。

3.1 マスキング機構のモデル化

マスキングモデルは CMLM の出力系列と最終層の隠れ状態を入力とした Transformer ベースの Decoder とする。これは CMLM の隠れ状態を source-target attention の source 側の情報として利用し、与えられた各トークンに対してマスクをかけるか否かを予測する Decoder である。このようなマスキング機構を用いれば出力トークンの尤度のような局所的な情報だけでなく、出力系列全体を考慮したマスキングが行える。また、後述する gumbel-softmax を利用した手法で推論時の挙動を再現した学習ができる。推論時には、予め設定した最大復号ステップに達するか、出力系列中の全トークンに対してマスクが必要ないと判定されれば復号終了とした。

3.2 復号一回のみの学習 (single-step)

学習開始時にいきなり複数回の復号を経た学習手法を用いると学習が安定しないため、まずは一回の復号に対して損失を計算する手法を提案する。この手法におけるトークン予測モデルの学習は、CMLM のものと同様に行う。マスク予測モデルについては、出力系列と教師系列を比較し、異なるトークンにはマスクをかけ、同一のトークンにはマスクをかけないような予測を行うよう、Cross-entropy 損失で学習させる。

3.3 複数回の復号を伴う学習 (multi-step)

学習時にも推論時の挙動を模倣させるため、複数回の復号ステップの後の最終的な出力に対する損失を復号過程全体に逆伝搬させる手法を提案する。Mask-Predict において、トークン予測モデルによって計算されたトークン出力分布は、その最大点 (argmax) を取る形でトークンに置き換えられる。このトークン選択は微分不可能な操作である。このトークンを選ぶ過程を擬似的に微分可能にするため、本研究では本来

argmax を取るところで gumbel-softmax によるサンプリングを利用する [5]。

Gumbel-softmax はカテゴリ分布から疑似的に微分可能な形でサンプリングする際に用いられる手法であり、式 (2) ような計算となる。

$$\text{gumbel-softmax}\left(p_i\right) = \frac{\exp\left(\left(\log p_i + g_i\right)/\tau\right)}{\sum_{j=1}^{I}\exp\left(\left(\log p_j + g_j\right)/\tau\right)} \tag{2}$$

ここで、p_i は語彙数 I のトークン予測モデルが出力するカテゴリ分布の i 番目の要素である。また、g_i は式 (3) ようにサンプリングされる gumbel ノイズである。温度パラメータ τ が小さいほど式 (2) の出力する分布は one-hot に近づく。

$$g_i = -\log\left(-\log u_i\right) \tag{3}$$

$$u_i = \text{uniform}\left(0, 1\right) \tag{4}$$

本研究の手法では、トークン予測モデルの出力系列から gumbel-softmax によるサンプリングを行い、得られたトークン列をマスク予測モデルの入力とする。同様にマスク予測モデルの出力系列についても gumbel-softmax でサンプリングし、得られたマスキング選択の系列を次の復号ステップでのマスキングに利用する。

微分不可能な選択操作を gumbel-softmax で代替することで、復号の全過程が擬似的に微分可能になるため、最後の復号ステップの出力に対する損失値をそれ以前の復号ステップにも逆伝搬させることができる。したがって、推論時の挙動を再現した学習が可能になる。

4 実験

4.1 実験データ

実験には WMT16 のルーマニア語-英語語対訳データ (wmt16 ro-en) [1] を用いた。データの大きさは学習用 600,000 文、開発用 1,999 文、テスト用 1,999 文である。データのトークン化は Lee ら [6] に習い、両言語の語彙を共有させた分割数 40,000 の BPE [7] で行った。

[1] https://www.statmt.org/wmt16/translation-task.html

4.2 学習設定

モデルや最適化のパラメータ設定は Ghazvininejad らの設定 [2] に従った。トークン予測モデルの Encoder と Decoder は埋め込み層・隠れ層の次元数を 512 とし、attention の head が 8 つの Transformer ブロックを 6 層を重ねたモデルとした。ただし、Transformer ブロックの Feedforward Network の次元数は Small CMLM に合わせ、512 とした。また、出力長予測モデルについては [3] の例に習い、出力長の絶対値ではなく翻訳元と翻訳後の系列長の差を予測する手法をとり、予測幅は $[-128, 128]$ に設定した。マスク予測の Decoder は attention の head が 8 つの Transformer ブロック 1 層のみのモデルとした。埋め込み層と隠れ層の次元数はトークン予測モデルと同様 512 とした。提案手法での gumbel-softmax の温度パラメータは 0.1 に固定した。ベースライン (Mask-Predict) のモデルは再現実装を行った。

4.3 結果

推論時は、出力長予測の候補数を 2、最大復号回数を 10 とした。出力文の例を表 1 に、BLEU スコアを表 2 に示す。BLEU スコアの計算には、開発データに対する BLEU スコアが高かった上位 5 つのチェックポイントのモデルパラメータを用い、開発・テストそれぞれについて 5 つのスコアの平均を取った。表 2 より、提案手法 (single-step) では従来の Mask-Predict と比較して若干の数値的な精度向上が見られたが、提案手法 (multi-step) では大幅に精度が悪化してしまった。いずれの手法でも復号にかかる時間はほぼ変わらず、Mask-Predict が僅かに遅い程度 (1.1 倍未満) だった。

5 考察

提案手法 (single-step) では従来の Mask-Predict と比較して若干の精度向上が見られた。ただし、いずれの手法も実験終了時にまだ学習が収束しきっていない可能性が考えられるため、提案手法 (single-step) 明確に従来の Mask-Predict よりも優れているとは言い切れない。出力結果についても、明確な差は見られなかった。

提案手法 (multi-step) では翻訳精度の大幅な劣化が見られた。これは、推論時の挙動を再現した学習手法が不安定であるからだと考えらる。gumbel-softmax

表 1: 出力文の例

入力文	“a fost o oportunitate pierduta in 2012”, a continuat fostul presedinte al finlandei.
Mask-Predict	“it was a lost opportunity in 2012,” former president of finland continued.
提案手法 (single-step)	“it was a lost opportunity in 2012,” the former finnish president continued.
提案手法 (multi-step)	“it was an missed lost in 2012,” the former president of finland.
正解文	“it was a missed opportunity in 2012,” continued the former president of finland.

表 2: 羅英翻訳の BLEU スコア (WMT16 ro-en)

モデル	dev	test
Mask-Predict (再現実装)	23.86	23.02
提案手法 (single-step)	24.74	24.09
提案手法 (multi-step)	14.12	13.94

はあくまで微分の近似値を計算可能にする手法であることから、必ずしも正確な微分値が求まるとは限らず、伝搬誤差の分散が大きくなり得る。また、適切な学習率等の実験設定も single-step のものとは異なる可能性がある。

提案手法の方がマスキング予測モデルの分だけパラメータの数が多いことを考えると、計算時間に大きな差異が見られなかったのは興味深い結果である。この原因として考えられるのは、Mask-Predict がマスキングにトークン予測の尤度を使う点である。尤度を基準としてトークンに順位をつける場合、比較を行う前に尤度を正規化する必要がある。このとき正規化に利用される softmax 関数は計算量が大きいことが知られている。Mask-Predict では復号する度に softmax 演算が必要になるため、計算量が増えてしまう。一方で提案手法では、モデルによってマスクをかける是非を決定する。モデルによるマスク予測では出力の尤度を正規化する必要がないため、softmax 演算を避けることができる。

出力結果については、いずれの手法でも同一トークンの繰り返しが見られたが、これについては提案手法 (single-step) の出力でより顕著だった。Mask-Predict と異なり、提案手法では復号の過程で明示的にマスクの数を減らす機構がない。特に翻訳が難しい長文に対しては、復号の後半になっても多数のマスクトークンが残ってしまっており、広域のトークンに対して予測を行っていた可能性が高い。提案手法 (multi-step) の学習が安定すれば、最終的な出力結果の評価が学習に反映されるためこの問題が解消される可能性も考えられる。

6 おわりに

本研究では、非自己回帰的機械翻訳における学習時と推論時の操作の違いに注目し、マスキング機構をモデル化することでそのギャップを埋める手法を提案した。翻訳の精度評価から、マスキング機能のモデル化による性能の劣化はほぼないことがわかった。一方で、推論時の復号過程を学習時にも再現した手法では学習が安定せず、翻訳精度が下がってしまった。

今後の課題としては、複数回の復号を行う学習を安定させることが挙げられる。また、本研究では複数回復号したあとの最終的な評価関数が Cross-entropy 損失によるものとなっているが、ここに強化学習などを適用して評価関数を BLEU スコアにするなどの発展も考えられる。

参考文献

[1] Bahdanau et al. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[2] Ghazvininejad et al. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*, 2019.

[3] Gu et al. Levenshtein transformer. In *NIPS*. 2019.

[4] Hochreiter et al. Long Short-Term Memory. *Neural Computation*, 9(8), 1997.

[5] Jang et al. Categorical Reparametrization with Gumble-Softmax. In *ICLR*, 2017.

[6] Lee et al. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *ACL*, 2018.

[7] Sennrich et al. Neural machine translation of rare words with subword units. In *ACL*, 2016.

[8] Vaswani et al. Attention is all you need. In *NIPS*. 2017.

[9] Zhang et al. Bridging the gap between training and inference for neural machine translation. In *ACL*, 2019.

[10] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.