単語出現頻度予測機能付き RNN エンコーダデコーダモデル

鈴木 潤　　永田 昌明
NTT コミュニケーション科学基礎研究所
{suzuki.jun, nagata.masaaki}@lab.ntt.co.jp

1 はじめに

リカレントニューラルネット (RNN) に基づくエンコーダデコーダモデルにより，例えば，機械翻訳 [18, 3]，質問応答 [22]，対話システム [20, 16]，文書要約 [15]，といった言語生成に関する自然言語処理タスクの性能が大幅に向上したという研究報告が多くみられるようになった．このモデルは，条件付き言語モデルと解釈できるので，言語生成タスクに適した方法論であると言える．しかし，より質の高い生成のために改善すべき課題はまだ多く残されている．一例として，同じフレーズ（単語）を何度も繰り返し生成する現象が散見されることが広く知られている．本稿では，この現象を便宜的に「繰り返し生成問題」と呼ぶことにする．

ニューラル翻訳の文脈では，被覆問題（coverage problem）の一部として，この繰り返し生成問題を間接的に軽減する方法論が既に提案されている [19, 11]．一方，翻訳以外の言語生成タスク，例えば，要約タスクでは，繰り返し生成問題を主題として議論している論文はみられない．しかし，要約タスクでは生成文の長さ制約を課す設定が一般的なので，繰り返し生成問題は，翻訳タスク以上に重大な解決すべき課題として取り扱うべき対象である．また，要約タスクにおいては，翻訳タスクで使われている被覆の考えや解決策をそのまま利用することができない．その理由は，要約のタスクの設定自体が，入力文から重要な要素を抽出して短くまとめた出力文を生成するタスクであるためである．つまり，入力文を被覆した出力文を生成するという考えが解きたいタスクそのものと相容れない概念となっている．例えば，文献 [12] では，翻訳タスクのカテゴリを「損失なし（loss-less）生成」と呼び，それと対比して，要約タスクを「損失あり圧縮（lossy compression）生成」と呼んでいる．このような背景から，本稿では，要約のような損失あり圧縮生成タスクにおいて，繰り返し生成問題を軽減する方法を提案する．

また，文献 [15] で提案された，要約タスクの一種であるヘッドライン生成タスクのデータを用いて提案法の有効性を検証する．

2 RNN エンコーダデコーダモデル

本稿のベースラインとなる RNN エンコーダデコーダモデルについて簡単に説明する．提案法のベースラインモデルは，文献 [10] で使われている注意機構（attention mechanism）付きのモデル [2] である．本モデルは，ニューラルネットに基づく翻訳や要約（ヘッドライン生成）タスクにおける強力なベースライン法という位置付けで，多くの論文で利用されている [4, 8]．モデルの設定には幾つかの選択肢があるが，本稿では，エンコーダとして「二層双方向 LSTM」，デコーダとして「注意機構付き二層 LSTM」を利用することとする．

本稿では，既に多くの参考文献が存在するので，スペースの都合でベースラインモデルの詳細説明は割愛する．ただし，以下に提案法を説明するのに必要な要素のみを述べる．まず，記号の煩雑化を避けるため，本稿では以下の 4 つの簡略記法を用いる．

1. 列ベクトルのリスト $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I)$ を，$(\boldsymbol{x}_i)_{i=1}^I$ と記述する．
2. ベクトルの要素が全て a である D 次元のベクトルを $\boldsymbol{v}(a, D)$ と記述する．
3. 記号 m は，いつでも出力側の語彙中の単語に一意に割り当てられた単語番号を表すこととする．$\mathcal{V}^{\mathrm{t}}$ を出力語彙（単語の集合）とすると，$m \in \{1, \ldots, |\mathcal{V}^t|\}$ である．ただし，以降 $M = |\mathcal{V}^t|$ とする．
4. $\boldsymbol{x}[i]$ は，ベクトル $\boldsymbol{x}$ の im 番目の要素を表す．

$\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^I$ を入力系列（入力文），$\boldsymbol{Y} = (\boldsymbol{y}_j)_{j=1}^J$ を出力系列（出力文）とする．ただし，$\boldsymbol{x}_i$ を i 番目の入力単語，$\boldsymbol{y}_j$ を j 番目の出力単語とする[*1]．

2.1 エンコーダ（符号化器）

$\Omega^{\mathrm{s}}(\cdot)$ を二層両方向 LSTM エンコーダの全ての処理を表す関数とする．エンコーダは，入力 $\boldsymbol{X}$ を受け取って隠れ状態ベクトルのリスト $\boldsymbol{H}^{\mathrm{s}} = (\boldsymbol{h}_i^{\mathrm{s}})_{i=1}^I$ を返す．

$$\boldsymbol{H}^{\mathrm{s}} = \Omega^{\mathrm{s}}(\boldsymbol{X}). \tag{1}$$

2.2 デコーダ（復号化器）

本稿では，K ベストビーム探索（beam-search）を用いて，入力系列 $\boldsymbol{X}$ が与えられたときの出現確率が最大となる出力系列 $\hat{\boldsymbol{Y}}$（の近似解）を獲得する．ビーム探索では，各処理時刻 j で K 個の出力候補を保持しながら探索を行う．Γ_K を K 個の（途中状態の）出力候補を選択する関数とする．このとき，処理時刻 j での K ベストビーム探索の処理は以下の式で記述できる．

$$\left\{(\hat{k}, \hat{m})_{k'}\right\}_{k'=1}^{K} = \Gamma^K \left\{\tilde{\boldsymbol{o}}_j^{(k)}[m] : \forall k, m\right\}. \tag{2}$$

[*1] 処理の単位は必ずしも単語である必要はないが，説明を簡単にするため，本稿では処理単位が単語と仮定して議論する．

スコアベクトル $\tilde{\boldsymbol{o}}_j^{(k)}$ は，以下の計算式で求められる．

$$\tilde{\boldsymbol{o}}_j^{(k)} = \boldsymbol{v}\Big(\tilde{\boldsymbol{o}}_{j-1}^{(k)}[m], M\Big) + \log\Big(\texttt{Softmax}(\boldsymbol{o}_j^{(k)})\Big)$$
$$\boldsymbol{o}_j^{(k)} = \Omega_j^{\mathrm{t}}(\boldsymbol{H}^{\mathrm{s}}, \hat{\boldsymbol{y}}_{j-1}^{(k)}), \tag{3}$$

ただし，$\texttt{Softmax}(\cdot)$ は，与えられたベクトルのソフトマックス（softmax）関数を表すとする．また，$\Omega_j^{\mathrm{t}}(\cdot)$ は，処理時刻 j でのデコーダの処理を表す関数とする．

3 単語頻度予測モデル

ここで，本稿の提案法を述べる．提案法は，大きく二つの構成要素からなる．

1. 出力系列に出現する各単語の頻度（の上限）を見積る補助モデル
2. 補助モデルの予測結果を用いてデコーダが選択する単語を制御する機構

3.1 定義

$\hat{\boldsymbol{a}}$ を出力語彙中の各単語の頻度予測結果のベクトル表記とする．$\odot$ を，要素単位の積を表す二項演算子とする．このとき，$\hat{\boldsymbol{a}}$ は以下の式で計算されると定義する．

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{r}} \odot \hat{\boldsymbol{g}}$$
$$\hat{\boldsymbol{r}} = \texttt{ReLU}(\boldsymbol{r}), \quad \hat{\boldsymbol{g}} = \texttt{Sigmoid}(\boldsymbol{g}), \tag{4}$$

ただし，$\texttt{Sigmoid}(\cdot)$ と $\texttt{ReLu}(\cdot)$ は，それぞれ，要素単位のシグモイド関数と ReLU 活性化関数 [5] とする．つまり，$\hat{\boldsymbol{r}} \in [0, +\infty]^M$，$\hat{\boldsymbol{g}} \in [0, 1]^M$，$\hat{\boldsymbol{a}} \in [0, +\infty]^M$ である．

提案法では，式 4 に示したとおり，二つの独立した補助モデル $\hat{\boldsymbol{r}}$ と $\hat{\boldsymbol{g}}$ を導入する．一つ目の $\hat{\boldsymbol{g}}$ は，単純に出力系列にその対象単語が出現するかしないかの確率モデルが返す確率に相当する．一方 $\hat{\boldsymbol{g}}$ は，対象単語の頻度（の上限）の予測値を表す．

本稿での単語出現確率モデル $\hat{\boldsymbol{g}}$ は，文献 [19] で被覆ベクトルを計算する際に利用する各単語毎の変換語数（fertility）の予測や，文献 [7] のコピー機構のスイッチ確率などを表現するために用いられるゲート関数と同じ発想で導入したモデルである．これは，単語頻度予測 $\hat{\boldsymbol{g}}$ の厳密な学習と予測は難しい問題なので，ゲート関数を導入することで，頻度 0 の場合はゲート関数が 0 を返すことで，$\hat{\boldsymbol{g}}$ の予測が上振れしても影響を与えないという効果が期待できる．

3.2 頻度予測結果の効率的な利用法

次に，頻度予測結果 $\hat{\boldsymbol{a}}$ の効率的な利用方法について議論する．$\hat{\boldsymbol{a}}$ の有効な利用法は幾つかの可能性が考えられる．本稿では，それら利用法の一つとして，デコード時の事前知識として利用する方法を提案する．ここで，式 3 中の $\tilde{\boldsymbol{o}}_j$ を以下のように再定義する[*2]．

$$\tilde{\boldsymbol{o}}_j = \boldsymbol{v}(\tilde{\boldsymbol{o}}_{j-1}[m], M) + \log\left(\texttt{Softmax}(\boldsymbol{o}_j)\right) + \tilde{\boldsymbol{a}}_j.$$

式 3 との違いは，$\tilde{\boldsymbol{a}}_j$ が追加された点である．$\tilde{\boldsymbol{a}}_j$ は，処理時刻 j で，元の $\hat{\boldsymbol{a}}_j$ を以下の式のように補正した尤度に対応する．

[*2] 可読性向上のため，ここでは k を破棄して記載する．

Input: $\boldsymbol{H}^{\mathrm{s}} = (\boldsymbol{h}_i^{\mathrm{s}})_{i=1}^{I}$ ▷ エンコーダ隠れ状態のリスト
Parameters: $\boldsymbol{W}_1^r, \boldsymbol{W}_1^g \in \mathbb{R}^{H\times H}$, $\boldsymbol{W}_2^r \in \mathbb{R}^{M\times H}$, $\boldsymbol{W}_2^g \in \mathbb{R}^{M\times 2H}$,
$\boldsymbol{H}_1^r \leftarrow \boldsymbol{W}_1^r \boldsymbol{H}^{\mathrm{s}}$ ▷ 頻度予測用の線形変換
$\boldsymbol{h}_1^r \leftarrow \boldsymbol{H}_1^r \boldsymbol{v}(1, M)$ ▷ $\boldsymbol{h}_1^r \in \mathbb{R}^H$, $\boldsymbol{H}_1^r \in \mathbb{R}^{H\times I}$
$\boldsymbol{r} \leftarrow \boldsymbol{W}_2^r \boldsymbol{h}_1^r$ ▷ 頻度推定結果
$\boldsymbol{H}_1^g \leftarrow \boldsymbol{W}_1^g \boldsymbol{H}^{\mathrm{s}}$ ▷ 出現予測用の線形変換
$\boldsymbol{h}_2^{g+} \leftarrow \texttt{RowWiseMax}(\boldsymbol{H}_1^g)$ ▷ $\boldsymbol{h}_2^{g+} \in \mathbb{R}^H$, $\boldsymbol{H}_1^g \in \mathbb{R}^{H\times I}$
$\boldsymbol{h}_2^{g-} \leftarrow \texttt{RowWiseMin}(\boldsymbol{H}_1^g)$ ▷ $\boldsymbol{h}_2^{g-} \in \mathbb{R}^H$, $\boldsymbol{H}_1^g \in \mathbb{R}^{H\times I}$
$\boldsymbol{g} \leftarrow \boldsymbol{W}_2^g(\texttt{concat}(\boldsymbol{h}_2^{g+}, \boldsymbol{h}_2^{g-}))$ ▷ 出現予測結果
Output: $(\boldsymbol{g}, \boldsymbol{r})$

図 1　単語出現頻度上限予測モデルの計算手順．

$$\tilde{\boldsymbol{a}}_j = \log\left(\texttt{ClipReLU}_1(\tilde{\boldsymbol{r}}_j) \odot \hat{\boldsymbol{g}}\right). \tag{5}$$

$\texttt{ClipReLU}_1(\cdot)$ は，$\boldsymbol{x}$ を受け取り $\boldsymbol{x}'$ を返す場合，全ての m に対して，$\boldsymbol{x}'[m] = \max(0, \min(1, \boldsymbol{x}[m]))$ となる計算を行う関数である．このとき，式 5 中の $\tilde{\boldsymbol{r}}_j$ と式 4 中の $\hat{\boldsymbol{r}}$ には，以下の関係が成り立つ．

$$\tilde{\boldsymbol{r}}_j = \begin{cases} \hat{\boldsymbol{r}} & \text{if } j = 1 \\ \tilde{\boldsymbol{r}}_{j-1} - \hat{\boldsymbol{y}}_{j-1} & \text{otherwise} \end{cases}. \tag{6}$$

式 6 は，一つ前の出力 $\hat{\boldsymbol{y}}_{j-1}$ を用いて，$\tilde{\boldsymbol{r}}_{j-1}$ から $\tilde{\boldsymbol{r}}_j$ に更新する．ここで，全ての j に対して $\hat{\boldsymbol{y}}_j \in \{0, 1\}^M$ なので，$\tilde{\boldsymbol{r}}_j$ 中の全ての要素は単調非増加である．もし，処理時刻 j のとき，$\tilde{\boldsymbol{r}}_{j'}[m] \leq 0$ なら，$\hat{\boldsymbol{s}}[m]$ によらず，$\tilde{\boldsymbol{o}}_{j'}[m] = -\infty$ となる．これの意味するところは，m 番目の単語は処理時刻 j 以降には絶対に出現しないことである．つまり，$\tilde{\boldsymbol{r}}_j$ は，出力単語が処理時刻 j 以降でどの程度出現できるかを表していると解釈できる．よって，提案法を用いた場合は，$\hat{\boldsymbol{r}}$ 以上の単語は出力系列に出現しないことを意味する．このような機構により，提案法では余計な繰り返し出力を抑制する．

ここで，一つ注意点として，提案法では，被覆の考え [19, 11, 21] のように，全ての m に対して $\tilde{\boldsymbol{r}}_j[m] \leq 0$（あるいは，$\tilde{\boldsymbol{r}}_j[m] = 0$）を満たすことを要件あるいはペナルティとしない．提案法で「（厳密な）出現頻度」の予測ではなく，「出現頻度の上限」の予測と記述しているのは，このためである．提案法では，冗長な出力を補正することが主たる目的なので，予測値を超えないことが重要であり，最終的に一致させることには必須の要件ではない．

3.3 計算方法

図 1 に，式 4 の $\boldsymbol{g}$ と $\boldsymbol{r}$ の計算手順を示す．まず，エンコーダが出すベクトルを全て足し合わせ，$\boldsymbol{r}$ を計算する．一方，出現確率予測モデル $\boldsymbol{g}$ に関しては，各出力単語が「出現するか」「しないか」に関する特徴を得たいので，出現する/しないに投票するような形式の特徴計算するようにした．例えば，ある入力単語が，ある出力単語が出現する/しないことに大きく影響をあたえる場合に，$\boldsymbol{g}$ は絶対値が大きな値を取ると考えられる．この考えと処理方式は，マックスプーリング（max pooling）やマックスアウト (MaxOut)[6] の考えを参考に導入した処理である．

3.4 パラメタ推定方法（学習）

$\boldsymbol{a}^* \in \mathbb{P}^M$ を正解の出現頻度のベクトル表記とする．ただし，$\mathbb{P} = \{0, 1, \ldots, +\infty\}$ とします．正解出現頻度 $\boldsymbol{a}^*$ は，エンコーダデコーダの学習に用いる正解学習

表 1　本稿で用いた学習および評価の設定．†: 特殊記号 BOS，EOS，UNK を含む．⋆: 文献 [21] に従う

Source vocabulary	† 119,507
Target vocabulary	† 68,887
Dim. of embedding D	200
Dim. of hidden state H	400
Encoder RNN unit	2 層双方向 LSTM
Decoder RNN unit	2 層 LSTM
Attention mechanism	あり
Dropout/Normalization	なし
Optimizer	⋆ Adam (最初の 5 epoch)[9] + SGD (残りの epoch)
初期学習率	0.001 (Adam) / 0.01 (SGD)
Mini batch size	256 (毎 epoch でシャッフル)
Gradient clipping	10 (Adam) / 5 (SGD)
終了判定条件	最大 15 epoch ただし開発データの精度に基づいて早期終了

データの出力系列から容易に獲得できる．次に，Ψ^{wfe} を単語出現頻度予測モデルを学習する際に用いる損失関数とします．

$$\Psi^{\mathrm{wfe}}\big(\boldsymbol{X}, \boldsymbol{a}^*, \mathcal{W}\big) = \boldsymbol{d} \cdot \boldsymbol{v}(1, M) \tag{7}$$
$$\boldsymbol{d} = c_1 \max\big(\boldsymbol{v}(0, M), \hat{\boldsymbol{a}} - \boldsymbol{a}^* - \boldsymbol{v}(\epsilon, M)\big)^b + c_2 \max\big(\boldsymbol{v}(0, M), \boldsymbol{a}^* - \hat{\boldsymbol{a}} - \boldsymbol{v}(\epsilon, M)\big)^b,$$

ただし，$\mathcal{W}$ は，エンコーダデコーダ中の全てのパラメタ行列（ベクトル）の集合とする．損失関数 $\Psi^{\mathrm{wfe}}(\cdot)$ は，サポートベクトル回帰（support vector regression: SVR）[17] の形と基本的に同じである．全ての m に対して $\hat{\boldsymbol{a}}[m]$ の値が，$[\boldsymbol{a}^*[m]-\epsilon, \boldsymbol{a}^*[m]+\epsilon]$ の範囲にある時，損失関数の値は 0 になる．提案法では，$\boldsymbol{a}^*$ が整数なので $\epsilon = 0.25$ を選択した．隣の整数とのマージンという考えに基づき両端の残りの 0.25 分を利用する．提案法では，$b = 2$ として，より予測を誤った場合に大きなペナルティを課すような設定を用いる．また，上限を見積るという考えを反映するために，$c_1 < c_2$ の設定を用いる．具体的には $c_1 = 0.2, c_2 = 1$ を用いる．

最終的に，式 7 の最小化問題をエンコーダデコーダの負の対数尤度最小化問題と線形結合し，同時に最適化を行うことで，ネットワーク中の全てのパラメタを推定する．

4　実験

文献 [15] で提案されたヘッドライン生成タスクのベンチマークデータを用いて提案法の有効性を検証する．このヘッドラインのベンチマークデータは，約 380 万文の訓練データ，40 万分の開発データ，40 万文のテストデータで構成される[*3]．ただし，実際に比較実験で利用する評価データは，評価データセクションからサンプリングされた 1951 文が通常用いられる．また，追加の評価データとして，DUC-2004 で利用された評価データ [14][*4]もよく用いられる．本稿では，ヘッドライン生成のベンチマークデータを用いて実験を行ってきた他の論文との比較を可能とするため，従来通りの実験設定を踏襲して実験を行った．表 1 に本稿で用いた実験の設定を一覧にして示す．

[*3] データ生成スクリプトは以下のサイトから入手可能 https://github.com/facebook/NAMAS.

[*4] http://duc.nist.gov/duc2004/tasks.html

表 2　実験結果（DUC-2004，Gigaword データ）

		DUC-2004 w/ 75-byte limit			Gigaword w/o length limit		
Method	Beam	R-1(R)	R-2(R)	R-L(R)	R-1(F)	R-2(F)	R-L(F)
EncDec	B=1	29.23	8.71	25.27	33.99	16.06	31.63
(baseline)	B=5	29.52	9.45	25.80	†34.27	†16.68	†32.14
	B=10	† 29.60	† 9.62	† 25.97	34.18	16.51	31.97
EncDec	B=1	31.92	9.36	27.22	36.21	16.87	33.55
+WFE	B=5	**⋆32.28**	**⋆10.54**	**⋆27.80**	**⋆36.30**	**⋆17.31**	**⋆33.88**
（提案法）	B=10	31.70	10.34	27.48	36.08	17.23	33.73
(† と ⋆ の差)		+2.68	+0.92	+1.83	+2.03	+0.63	+1.78

G: china success at youth world championship shows preparation for #### olympics
A: china germany germany germany germany and germany at world youth championship
B: china faces germany at world youth championship

G: British and Spanish governments leave extradition of Pinochet to courts
A: spain britain seek shelter from pinochet 's pinochet case over pinochet 's
B: spain britain seek shelter over pinochet 's possible extradition from spain

G: torn UNK : plum island juniper duo now just a lone tree
A: black women black women black in black code
B: in plum island of the ancient

図 2　生成例．G: 正解要約 A: ベースラインエンコーダデコーダ B: 提案法 (下線部は冗長な繰り返し生成部分)

表 3　現在のトップシステムとの比較．‘*’: これまでの最良の結果．†: 実験の設定が異なるため参考結果

	DUC-2004			Gigaword		
Method	R-1(R)	R-2(R)	R-L(R)	R-1(F)	R-2(F)	R-L(F)
ABS [15]	26.55	7.06	22.05	30.88	12.22	27.77
RAS [4]	28.97	8.26	24.06	33.78	15.97	31.15
BWL [12][*5]	28.35	9.46	24.59	32.67	15.59	30.64
(words-lvt5k-1sent†)	28.61	9.42	25.24	35.30	*16.64	32.62
MRT [1]	*30.41	***10.87**	*26.79	***36.54**	16.59	*33.44
EncDec+WFE	**32.28**	10.54	**27.80**	36.30	**17.31**	**33.88**

4.1　ベースラインとの比較実験結果

表 2 に比較実験の結果を示す．**R-1(R)**，**R-2(R)**，**R-L(R)** は，それぞれ再現率ベースの ROUGE-1，ROUGE-2，ROUGE-L スコアである．同様に，**R-1(F)**，**R-2(F)**，**R-L(F)** はそれぞれ F 値ベースの ROUGE-1，ROUGE-2，ROUGE-L スコアである．本実験のベースラインの立ち位置を定量的に測るために，`OpenNMT` ツール[*6]を用い，本実験のベースラインと同じ設定で実験をおこなった．その結果 Gigaword データで $B = 5$ の時，それぞれ R-1(F) が 33.65，R-2(F) が)16.12，R-L(F) が 31.37 であった．これは，表 3 に示すベースラインの結果とほぼ同等（若干下回る）であることから，本実験でのベースラインは非常に良い精度を達成していると言える．

また，提案法がベースラインを大きく上回る結果が得られていることがわかる．ベースラインと提案法の違いは，単語出現頻度上限を予測する補助モデルがあるかないかの違いである．このことから，単語出現頻度上限を予測する補助モデルが性能向上に大きく貢献

[*6] https://github.com/harvardnlp/seq2seq-attn

していることが示せた．

4.2 生成例

図 2 に実際の生成例を示す．本稿の動機にあたる冗長な繰り返し生成を削減できているかの部分を定性的に評価した．図からわかるように，提案法はベースラインと比較して繰り返しを大幅に削減できていることが見て取れる．特に人間の主観評価では，こういったあからさまな間違いは大きな悪印象を与えることなる場合が多いので，その観点では，提案法は大きく貢献していると言える．

4.3 現在のトップシステムとの精度比較

表 3 に現在のトップシステムとの精度比較を示す．表からわかるように，提案法は，自動評価指標で現在最もよい結果を上回る結果が得られた．

これまで最良の結果を得ていたのは MRT[1] である．この方法と提案法のベースラインモデルは，ほぼ同じと考えられる．ただし，MRT では，系列全体の最小リスク推定によりパラメタ更新を行う方法論が主要な主張点である．MRT で鍵となる最小リスク推定に基づくパラメタ推定法は，もちろん提案法でも利用することが可能である．よって，提案法と MRT で持ちられるパラメタ推定法を組み合わせることで，更なる精度向上が期待できる．

5 おわりに

本稿では，要約のような損失あり圧縮生成タスクに対して，冗長な繰り返し生成を抑制することを主たる目的とした方法を提案した．提案法は，出力系列に出現すると思われる単語頻度の上限をあらかじめ予測し，予測した頻度以上の単語出現を制限することで，結果として冗長な繰り返し生成を抑える方法である．実験では，ヘッドライン生成のベンチマークデータを用いて，提案法の効果を調査した．提案法は，定量評価で，これまでで最も高い精度を達成した．提案法は，要約にかぎらず，例えば，画像からのキャプション生成タスクのように他の損失あり圧縮生成タスクでも有効に働くことが期待できる．

参考文献

[1] Ayana, Shiqi Shen, Zhiyuan Liu, and Maosong Sun. Neural headline generation with minimum risk training. *CoRR*, Vol. abs/1604.01904, , 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2014.

[3] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1724–1734, 2014.

[4] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In Geoffrey J. Gordon and David B. Dunson, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, Vol. 15, pp. 315–323. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011.

[6] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, and Yoshua Bengio. Maxout Networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pp. 1319–1327, 2013.

[7] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 140–149, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, Austin, Texas, November 2016. Association for Computational Linguistics.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, Vol. abs/1412.6980, , 2014.

[10] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[11] Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 955–960, Austin, Texas, November 2016. Association for Computational Linguistics.

[12] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, Vol. abs/1602.06023, , 2016.

[13] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[14] Paul Over, Hoa Dang, and Donna Harman. DUC in context. *Information Processing and Management*, Vol. 43, No. 6, pp. 1506–1520, 2007.

[15] Alexander M. Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 379–389, 2015.

[16] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural Responding Machine for Short-Text Conversation. *CoRR*, Vol. abs/1503.02364, , 2015.

[17] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, Vol. 14, No. 3, pp. 199–222, 2004.

[18] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 3104–3112, 2014.

[19] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics.

[20] Oriol Vinyals and Quoc V. Le. A Neural Conversational Model. *CoRR*, Vol. abs/1506.05869, , 2015.

[21] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, Vol. abs/1609.08144, , 2016.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 2048–2057, 2015.