Non-Factoid 型質問のための結論と理由で構成される回答文の生成手法

Conclusion-Supplement Answer Generation for Non-Factoid Questions

中辻 真
Makoto Nakatsuji

NTT レゾナント株式会社
nakatsuji@nttr.co.jp

八島 浩文
Hirofumi Yashima

(同上)
y-hiro@nttr.co.jp

keywords: non-factoid QA, answer generation, encoder-decoder model, ensemble network

Summary

This paper tackles the goal of conclusion-supplement answer generation for non-factoid questions, which is a critical issue in the field of Natural Language Processing (NLP) and Artificial Intelligence (AI), as users often require supplementary information before accepting a conclusion. The current encoder-decoder framework, however, has difficulty generating such answers, since it may become confused when it tries to learn several different long answers to the same non-factoid question. Our solution, called an *ensemble network*, goes beyond single short sentences and fuses logically connected conclusion statements and supplementary statements. It extracts *the context* from the conclusion decoder's output sequence and uses it to create supplementary decoder states on the basis of an attention mechanism. It also assesses *the closeness* of the question encoder's output sequence and the separate outputs of the conclusion and supplement decoders as well as *their combination*. As a result, it generates answers that match the questions and have natural-sounding supplementary sequences in line with the context expressed by the conclusion sequence. Evaluations conducted on datasets including "Love Advice" and "Arts & Humanities" categories indicate that our model outputs much more accurate results than the tested baseline models do.

1. はじめに

質問回答 (QA) システムは, Apple の Siri や Amazon の Echo などの近年の対話を基にした自然言語理解システムにおいて重要な役割を果たしている. ユーザは, 求めている答えを得るために, AI システムと会話をすることになってきている. QA システムは, 大きく, 事実型 (Factoid) と非事実型 (Non-Factoid) に分類される. 前者は, 「*X* は何/誰ですか?」のように, 物や人に対する事実を質問するものである. 後者はより多様な質問が含まれ, 短い事実を引用するだけでは回答できず, 回答に対する根拠や事例を必要とするなど, より長文となる傾向がある. 例えば, 遠距離恋愛を成功させる方法や, 社会的な争点となっている問題などに対する質問文などが該当する. Factoid 型質問に回答する研究は, 近年, 大きな進展がみられている [Wang 07, Yu 14] が, Non-Factoid 型質問に精度良く答えることは, 現在も困難である.

対話システムによく応用されている Long Short Term Memory (LSTM) に基づく Sequence-to-sequence (Seq2seq) モデル [Sutskever 14, Vinyals 15, Bahdanau 14] は, 短い発話を入力として, 短い回答文を生成できる. そして, これらのモデルは, 単純な形式の Factoid 型質問には対応

できる可能性を持つことが検証で示されている [Vinyals 15]. 例えば, 「空は青いですか? それとも黒いですか?」といった常識を問う質問文に対しては, 学習用コーパスの中で頻出する文章を学習することで, 回答を生成できる. また, 近年の機械読解手法 [Nguyen 16, Rajpurkar 16] は, 直前の質問文から回答文字列を抜粋することで, 単一の短文の回答文を生成する課題に挑戦している. しかし, [Jia 17, Wang 18, Nakatsuji 20] で指摘されているように, この手法では, Non-Factoid 型質問のように, 一つの質問に対し複数の異なる結論や根拠を含む長文回答が存在する場合には, 学習に混乱を生じ, 満足のいく回答文を生成できない.

本稿では, Non-Factoid 型質問に対する回答生成を実現するため, 上述した現在の文章生成の課題を踏まえ, 結論部と補足部の組合せから構成される回答文を生成するという新たな問題を設定し, 本問題の解決に挑戦する [Nakatsuji 20b]. ここで, 結論部は, 質問文に端的に回答する文で構成され, 補足部は結論部を裏付ける, 結論に至る理由や事例からなる文で構成される. こうした, 結論部-補足部で構成される回答文は, 質問者が自らの今後の行動を決める上での助けとなる. このことは, 批判的

思考に関する研究 [Ennis 91] でも述べられており、ユーザは、提示された行動指示（結論）を受け入れ行動する前に、結論を裏付ける補足情報を受け取りたいと考えているとされる。また、ディベートにおける研究でも述べられており、優れた討論者は、論題に対する主張（結論等）に加え、ユーザが意思決定をするのを促すための補足情報を提示するものである [Rinott 15]。

結論部-補足部からなる回答を例を用い説明する。例えば、「遠距離恋愛は困難ですか？」というユーザからの質問に対し、既存手法では、「真の愛情は遠距離恋愛を克服する」といったような短文の回答文が生成され得る。しかし、質問者はこのような単純な回答文では満足できないことが多い。なぜならば、質問者は、そうした結論に至る理由を知りたいと考えているからである。もしも、「距離は愛情が本物かどうかを試す」といった補足情報が結論に添えて提示されたならば、質問者は結論を受け入れることができ、今後の意思決定に AI システムから提示された結論を活用することができる。また、Non-Factoid 型の質問文の中には、複数の回答文が存在するものもある。例えば先ほどの質問文に対する回答文としては、次の文も候補となり得る。「遠距離関係は、たいていの人間関係を疎遠にさせるものである。あなたは相手と密に連絡を取り合うべきである」。しかしながら、既存手法は、上述したように、同一の質問に複数の異なる回答文が存在する場合、学習に混乱をきたしてしまう。そのため、本稿の取り扱う、結論部と補足部で構成される回答文の生成にそのまま適用することは困難である。

上述した課題を解決するため、我々はアンサンブル・ネットワークと名付けた新しいアーキテクチャを提案する。提案手法は、既存のエンコーダ・デコーダモデルの拡張モデルであり、デコーダ部は、結論部と補足部の 2 つの異なる文を、以下の 2 つの視点から生成する。(1) デコーダから出力される結論部の系列を、補足部に関連付ける。このようにして、結論部が持つ文脈を自然に引き継いで、補足部は生成される。(2) 質問文と結論部の類似性および、質問文と補足部の類似性を最適化すると同時に、結論部と補足部の間の組合せの最適化も行う。特に、単語レベルだけでなく、文間の類似性と組合せを評価することで、結論部の文脈に即した補足部を含む回答文を生成できる。これにより、同じ Non-Factoid 型質問に対応する複数の結論部と補足部の多様な組合せを学習する必要性を回避することができ、論理的に整合の取れた結論と補足を含む回答文を生成できる。

コミュニティベースの QA (Community QA, CQA) サイトは、結論部と補足部によって構成される回答を提示する傾向がある。例えば、我々が調査したところ、教えて!goo^{*1}の Non-Factoid 型質問（恋愛相談）や、Yahoo! Answers の Non-Factoid 型質問のデータセット^{*2}におけ

る「社会科学」、「文化芸術」、「人文科学」領域では、多くの回答が、結論部と補足部から構成されている。また、上記の CQA データセットは、多くの回答者によって作成された多様な回答文が含まれる。そのため、我々の目的とする Non-Factoid 型の回答生成の検証に適切であるため、提案モデルの評価では、これらのデータセットを用いた。結果、我々の提案モデルが、既存手法よりも、正確かつ自然な回答文を生成することが確認できた。

2. 関 連 研 究

エンコーダ・デコーダフレームワークは、1 つの表現系列を、別の表現系列へと変換する方法を学習するフレームワークである。例えば、Contextual LSTM (CLSTM) は、文脈が持つ特徴（トピック）を、本フレームワークに取り入れている [Ghosh 16]。これは、質問文が持つ文脈を、回答文を生成する際に活用できる。階層型エンコーダ・デコーダモデル (HRED) [Serban 16] は、階層型リカレント・エンコーダ・デコーダモデルを、会話領域に利用可能な形に拡張させたものである。HRED は、質問文を密度の高い文脈ベクトルに圧縮した後に、複数の回答文を再帰的にデコードし生成可能である。しかしながら、回答文を構成する次の単語トークンを順次生成していく手法は、回答文が長くなるほど、質問文がもともと持っていた本来の意味を喪失し破綻した回答を生成することが多い。また、近年では、パッセージランキングモデル [Nogueira 19, Liu 18] などの Transformer [Vaswani 17] に基づく手法が、QA モデルを評価するために提案されている。ただし、Non-Factoid 型の回答文を生成するモデルとして、Transformer ベースのものは現状ほとんどない。

一方、近年提案された Non-Factoid 型質問のためのニューラル回答選択モデル [Santos 15, Qiu 15, Tan 16, 中辻 19] は、質問文と回答文の特徴表現を学習し、Cosine 類似度に基づく類似性計測指標を用い、両者のマッチングを学習する。これらの研究は、Yahoo! Answers などの CQA ウェブサイトに保存されているオープンデータセットを用いている。しかしながら、これらのモデルは、回答文を、選択し抽出するのみであり、生成はしていない。

また、近年の機械読解モデルは、与えられた文章（パッセージ）から文字列を抜粋することで、質問文に対する回答の生成を試みている [Yu 18, Rajpurkar 16, Yang 15, Joshi 17, Nakatsuji 20]。例えば、MS-MARCO データセットを用いた先行研究 [Tan 17, Nguyen 16, Wang 18, Nakatsuji 20] では、与えられた複数のパッセージ中に存在しない単語から構成される回答文の生成をも試みている。しかし、これらのモデルは、学習用および検証用データセットとして、QA ペア以外にパッセージを準備する必要がある。そのためパッセージが存在しない CQA データセットに対しては、適用できない。さらに、デー

*1 <https://oshiete.goo.ne.jp>

*2 <https://ciir.cs.umass.edu/downloads/nfl6/>

図1 提案モデルのアーキテクチャ概要.

タセット中の質問文の多くは単一の回答文しか持たない。そのため、機械読解の先行研究が目指しているところは、我々の目的とする多様な回答文が求められる Non-Factoid 型の回答生成とは異なる。

3. 提案手法

本章では、提案手法について、詳しく説明する。手法のアーキテクチャを、図1に示す。

提案モデルは、入力として質問 $Q = \{q_1, \dots, q_i, \dots, q_{N_q}\}$ を受け取り、結論部 $C = \{c_1, \dots, c_t, \dots, c_{N_c}\}$ および、補足部 $S = \{s_1, \dots, s_t, \dots, s_{N_s}\}$ を生成する。

ここで、 q_i は、長さ N_q を持つ入力 Q 中の i 番目の単語の単語ベクトル、 c_t (または s_t) は、長さ N_c (または N_s) を持つ出力 C (または S) 中の t 番目の単語の単語ベクトルを意味する。

3.1 エンコーダ

エンコーダは、入力文 Q を、質問ベクトル O_q と潜在ベクトル $H = \{h_i\}_i$ に変換する。

質問文は、質問に関するいくつかの背景情報 (例: 質問者が置かれた状況) が質問内容そのものとともに含まれているため、長文となりがちであり、複数の文から構成されることが多い。このため、我々は、質問文を双方向からエンコードすることで、質問文が持つ意味内容をよりよく捉えることができる BiLSTM エンコーダを用いる。そして各時点 i に対し、エンコーダは潜在ベクトルの値を以下のように更新する。

$$h_i = [h_i^f, h_i^b]^T \text{ s.t.}$$

$$h_i^f = f(q_i, h_{i-1}^f), h_i^b = f(q_i, h_{i+1}^b),$$

ここで、 $f()$ は LSTM のユニットであり、 h_i^f と h_i^b はそれぞれ、順方向の LSTM と逆方向の LSTM から出力された潜在ベクトルの状態である。

我々は、モデルが結論部や補足部をよりよく理解できるようにするために、結論部や補足部が持つ文としての型の特徴を、系列変換モデルを学習する過程に反映させ

たいと考えている。そのため、結論 C または補足 S を示す文型ベクトルを定義し、それらを、LSTM モデルの中の入力ゲート、忘却ゲート、およびセルメモリに追加する。この方法は、LSTM モデルの中で、文の特徴ベクトルと文型特徴ベクトルを結合させる複合入力 $[q_i, C]$ または $[q_i, S]$ を処理するのと同義である。我々は、この改変された LSTM を上述の BiLSTM モデルで以下のように用いる。

$$h_i = [h_i^f, h_i^b]^T \text{ s.t.}$$

$$h_i^f = f([q_i, C], h_{i-1}^f), h_i^b = f([q_i, S], h_{i+1}^b).$$

なお補足部の系列文をデコードするために、質問文をエンコードする場合、上記の式中の C の代わりに S が入力される。 C および S は、後述する損失関数に基づく学習対象であり、初期値はランダムに生成される。

次に、BiLSTM エンコーダは、すべての潜在ベクトルにマックス・プーリングを適用して、各単語を出力する際に意味合いの大きいベクトル値を抽出する。本稿におけるマックス・プーリングは、結論部の計算を例とすると、入力される潜在ベクトル列 $\{h_1, \dots, h_i, \dots, h_{N_q}\}$ に対し、各ベクトルの j 番目の要素の $\max$ 値を、出力となるベクトル O_q^c の j 番目の要素とし、 $O_{qj}^c = \max\{h_{ij}\}_{i=1}^{N_q}$ のように計算される (なお、本稿における後段で登場するマックス・プーリングはすべて同様の処理である)。その結果、結論部用の固定サイズの分散ベクトル表現 O_q^c と、補足部用の別の分散ベクトル表現 O_q^s が生成される。 O_q^c と O_q^s は、対応する文型ベクトル C または S によってバイアスされているため互いに異なる値を持つ。

図1の例では、BiLSTM エンコーダは、各単語を文型ベクトル (C または S) とともに処理し、マックス・プーリングを適用することで、質問ベクトル O_q^c または O_q^s を生成する。これらの特徴ベクトルは、結論部と補足部のデコーダにおいて、文脈ベクトルとして適用される。

3.2 デコーダ

デコーダは、結論部のデコーダと補足部のデコーダで構成される。

エンコーダと同様に、デコーダも、結論における単語ベクトルと文型を示すベクトルを連結する LSTM セル内の複入力 $[\mathbf{c}_t, \mathbf{C}]$ をデコードし、結論デコーダの LSTM ユニットの t 番目の潜在ベクトル $\mathbf{h}_t^c$ は、以下の式で計算される。なお、結論部のデコーダには、 $\mathbf{O}_q^c$ が、デコーダの先頭、つまり、 $\mathbf{c}_0$ としてセットされ、また、補足部のデコーダには、 $\mathbf{O}_q^s$ が、デコーダの先頭、つまり、 $\mathbf{s}_0$ としてセットされる。

$$\begin{aligned}\mathbf{h}_t^c &= f'([\mathbf{c}_{t-1}, \mathbf{C}], \mathbf{h}_{t-1}^c) \text{ s.t.} \\ \mathbf{c}_{t-1} &= \underset{c}{\operatorname{argmax}} \underset{c}{\operatorname{softmax}}(\mathbf{h}_{t-1}^c).\end{aligned}$$

ここで、 $f'()$ は、結論部をデコードする LSTM セルを意味し、 $\operatorname{softmax}_c$ は、 $\operatorname{softmax}$ 層によって与えられる単語 c の確率を意味し、 c_t は、デコードされた結論部の文中の t 番目のトークンを、さらに $\mathbf{c}_t$ は c_t の単語ベクトルを意味する。補足部のデコーダの潜在ベクトル $\mathbf{h}_t^s$ は、 $\mathbf{h}_t^c$ と同様の方法で計算される。ただし、 $\mathbf{h}_t^s$ は、次節で説明するアンサンブルネットワークで更新される。

図 1 の例では、LSTM デコーダは、 $\mathbf{O}_q^c$ または $\mathbf{O}_q^s$ をエンコーダから受け取り、各時刻 t の潜在ベクトルを生成する。次に、これらの出力が次に述べるアンサンブル・ネットワークに入力される。

3.3 アンサンブル・ネットワーク

従来のエンコーダ・デコーダフレームワークは、Non-Factoid 型質問に対して適切に答えることができない短文で単純な文を生成する傾向にある。長い回答文を生成するように強制しても、デコーダから出力される文系列は、文の始めから終りにかけて、一貫性を欠くものになってしまう。

アンサンブル・ネットワークはこの課題を、(1) 結論部のデコーダから出力された系列の文脈を、補足部のデコーダに渡し、(2) エンコーダへの入力文とデコーダの出力文との近さと、デコーダから出力された文間の組合せを考慮することによって解決する。これにより、同じ Non-Factoid 型質問に対応する複数の結論部と補足部の多様な組合せを学習する際の混乱を回避することができ、論理的に整合の取れた結論と補足を含む回答文を生成できる。

以下に上記 2 つのアプローチをそれぞれ詳説する。

(1) 文脈を制御するために、結論部のデコーダから出力された潜在ベクトルを評価し、結論ベクトル $\mathbf{O}_c$ を計算する。そのため、BiLSTM を結論部のデコーダの出力状態 $\{\tilde{\mathbf{y}}_t^c\}_t = \{\mathbf{U} \cdot \operatorname{softmax}(\mathbf{h}_t^c)\}_t$ に適用する。ここで、単語表現行列 $\mathbf{U}$ は、その列に単語表現を保持するものである。

そして、BiLSTM エンコーダは、時点 t における、潜在ベクトルの値を次式で更新する。

$$\begin{aligned}\mathbf{h}_t^c &= [\mathbf{h}_t^{c,f}, \mathbf{h}_t^{c,b}]^T \text{ s.t.} \\ \mathbf{h}_t^{c,f} &= f(\tilde{\mathbf{y}}_t^c, \mathbf{h}_{t-1}^{c,f}), \quad \mathbf{h}_t^{c,b} = f(\tilde{\mathbf{y}}_t^c, \mathbf{h}_{t+1}^{c,b}),\end{aligned}$$

ここで、 $\mathbf{h}_t^{c,f}$ および $\mathbf{h}_t^{c,b}$ は、結論部のエンコーダにある前方向 LSTM と後ろ方向 LSTM から出力される潜在ベクトルである。すべての潜在ベクトルにマックス・プーリングを適用することで、結論ベクトル $\mathbf{O}_c$ を生成する。次に、時点 t の文脈ベクトル $\mathbf{c}\mathbf{x}_t$ を、補足部を出力するデコーダの $(t-1)$ 番目の重み行列 $\mathbf{h}_{t-1}^s$, $\mathbf{V}_a$, $\mathbf{W}_a$ およびシングモイド関数 σ を用いて以下のように計算する。

$$\mathbf{c}\mathbf{x}_t = \alpha_t \mathbf{O}_c \text{ s.t. } \alpha_t = \sigma(\mathbf{V}_a^T \tanh(\mathbf{W}_a \mathbf{h}_{t-1}^s + \mathbf{O}_c)).$$

アンサンブル・ネットワークは、この計算により得られた $\mathbf{c}\mathbf{x}_t$ を、結論部の文レベルのコンテキストとして、補足部の単語生成時に以下の式 (1) の形式で与える。結果として得られる補足部は、結論部の文脈に沿うものになる。最後に、 $\mathbf{h}_t^s$ は次のように計算される。

$$\begin{aligned}\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{y}_{t-1}, \mathbf{T}] + \mathbf{U}_z \mathbf{h}_{t-1}^s + \mathbf{W}_z^a \mathbf{c}\mathbf{x}_t + \mathbf{b}_z) \quad (1) \\ \tilde{\mathbf{l}}_t &= \tanh(\mathbf{W}_l[\mathbf{y}_{t-1}, \mathbf{T}] + \mathbf{U}_l \mathbf{h}_{t-1}^s + \mathbf{W}_l^a \mathbf{c}\mathbf{x}_t + \mathbf{b}_l) \\ \mathbf{l}_t &= \mathbf{i}_t * \tilde{\mathbf{l}}_t + \mathbf{f}_t * \mathbf{l}_{t-1} \\ \mathbf{h}_t^s &= \mathbf{o}_t * \tanh(\mathbf{l}_t).\end{aligned}$$

ここで z は、それぞれ、3 つのゲート（入力 i 、忘却 f 、および出力 o ）を取り得る変数であり、その時点 t の潜在ベクトルは、 $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ になり得る。 $\mathbf{l}_t$ は、セル・メモリベクタを意味する。 $\mathbf{W}_z^a$ および $\mathbf{W}_l^a$ は、パラメータである。

(2) 文の近さと文の組合せを制御するために、補足部のデコーダ部から出力されたすべての情報を評価し、補足ベクトル $\mathbf{O}_s$ を $\mathbf{O}_c$ と同様にして計算する。つまり、BiLSTM を補足部のデコーダ部からの出力状態 $\{\tilde{\mathbf{y}}_t^s\}_t = \{\mathbf{U} \cdot \operatorname{softmax}(\mathbf{h}_t^s)\}_t$ に対して適用し、さらにマックス・プーリングを用いて $\mathbf{O}_s$ を計算する。最後に、結論部と補足部で構成される回答文を生成するために、質問ベクトル $\mathbf{O}_q$ と回答文ベクトル ($\mathbf{O}_c$ または $\mathbf{O}_s$) との間の近さおよび $\mathbf{O}_c$ と $\mathbf{O}_s$ の組合せを最適化する。この最適化の計測指標の損失関数については、次節で説明する。

図 1 に示すように、アンサンブル・ネットワークは、結論ベクトル $\mathbf{O}_c$ と、結論部のデコーダからの補足ベクトルへの重み値および補足ベクトル $\mathbf{O}_s$ を計算する。そして、 $\mathbf{O}_c$ と $\mathbf{O}_s$ は、質問ベクトル ($\mathbf{O}_q = [\mathbf{O}_q^c, \mathbf{O}_q^s]$) とともにアンサンブル・ネットワークの損失関数に入力される。

3.4 アンサンブル・ネットワークの損失関数

提案モデルでは、 $p(\mathbf{y}|\mathbf{q})$ を生成する条件付き確率を最大化することを目的とする従来のエンコーダ・デコーダ

モデルの損失関数をそのまま用いることはしない。これは、質問文と回答文の近さと、2つの回答文系列に対する組合せを評価することが、自然な回答を生成するのに役立つと考えるためである。

損失関数は、質問文と結論部の近さ、および質問文と補足部の近さを最適化し、さらに結論部と補足部の組合せを最適化する。学習時の損失値 $\mathcal{L}_s$ は、次のヒンジ損失として表される。ここで、 $\mathbf{O}^+$ は、正解の出力ベクトル、 $\mathbf{O}^-$ は、回答空間全体からランダムに選択された不正解の回答文、 M は定数マージンであり、 $\mathbf{A}$ は結論と補足の組合せの負例集合 $\{[\mathbf{O}_c^+, \mathbf{O}_s^-], [\mathbf{O}_c^-, \mathbf{O}_s^+], [\mathbf{O}_c^-, \mathbf{O}_s^-]\}$ である。

$$\mathcal{L}_s = \sum_{\mathbf{O}_a \in \mathbf{A}} \max\{0, M - (\cos(\mathbf{O}_q, [\mathbf{O}_c^+, \mathbf{O}_s^+]) - \cos(\mathbf{O}_q, \mathbf{O}_a))\}.$$

本式において $\mathcal{L}_s$ は、結論部、補足部、およびそれらの組合せが適切に予測されているかどうかを評価している。そうすることで $\mathcal{L}_s$ は、結論部や補足部の予測だけでなく、結論部と補足部の組合せの予測までも最適化できる。

図1の上部に示すように $(\mathbf{O}_q, \mathbf{O}_c, \mathbf{O}_s)$ は、デコーダの出力系列の質問ベクトルとの近さと系列間の組合せを最適化するため、アンサンブル・ネットワークに入力されている。

3.5 学 習

学習損失 $\mathcal{L}_w$ は、 $\mathcal{L}_s$ とエンコーダ・デコーダモデルのクロスエントロピー損失の両方から構成されている。次の式では、結論部と補足部の系列文が、長さ T の1つの系列 $\mathbf{Y}$ にマージされる。ここで、 $T = N_c + N_s$ である。

$$\mathcal{L}_w = \alpha \cdot \mathcal{L}_s - \ln \prod_{t=1}^T p(\mathbf{y}_t | \mathbf{Q}, \mathbf{y}_1, \dots, \mathbf{y}_{t-1}). \quad (2)$$

ここで α は、2つの損失の重み付けを制御するためのパラメータである。提案手法では適応確率的勾配降下法 (AdaGrad) を使用して、モデルを学習する。

図1においても、アンサンブル・ネットワークの損失関数と、クロスエントロピーの損失を図示している。

なお、結論部デコーダと補足部デコーダ、および、アンサンブル・ネットワークにおける結論部エンコーダと補足部エンコーダは、メモリ効率化と少ない学習データにおける文章生成効率の向上を目的にパラメータを共有する。一方、質問エンコーダと結論部エンコーダおよび、質問エンコーダと補足部エンコーダは、質問と回答部は明らかに文章の性質が異なることから、パラメータは別に用意し、学習を行う。

4. 評 価 検 証

本章では、提案手法の検証について説明する。

4.1 比 較 手 法

検証では、提案手法 *NAGM* を以下の5つの手法と比較した。

- *Seq2seq*: [Bahdanau 14] によって提案された Seq2seq アテンション・モデル。
- *CLSTM*: [Ghosh 16] によって提案された CLSTM モデル。
- *Trans*: [Vaswani 17] により提案され、自然言語処理領域において汎用的に有効であると証明された Transformer モデル。
- *HRED*: 過去のエンコード結果を文脈として引継ぎながら逐次的に次の文をデコードする階層型リカレント・エンコーダ・デコーダモデル [Serban 16] を、結論部と補足部の文脈を学習し、補足部を結論部を受けて出力するように適用したモデル。
- *NAGMWA*: 結論から補足への文脈を取り除いた提案手法。具体的には、式 (1) にある $\mathbf{cx}_t$ をデコーダへ橋渡ししない。デコーダから出力される結論文と補足文は、損失関数 $\mathcal{L}_s$ によって結合される。

Seq2seq, *CLSTM*, *Trans* は、結論部の系列文と補足部の系列文を順番にデコードする方法よりも精度が高いことが事前検証で確認できたため、結論部の系列文と補足部の系列文を別々にエンコーダ・デコーダの組を2つ用意しそれぞれデコードした後に、両者を結合することで回答文を生成する。なお、これらの手法については、我々の提案手法とは異なり、文型ベクトルや、文と文の組合せ学習は提案されていないため用いていない。また、質問エンコーダを共通に、結論と補足のデコーダを別々に用意するという方法は、エンコーダ・デコーダの組を2つ作る上述の方法よりも、結論、補足に最適化された学習ができないため採用していない。

4.2 デ ー タ セ ャ ッ ト

検証では、以下の2つの CQA データセットを使用した。

i. 教えて!goo

教えて!goo データセットには、日本語の CQA サイト「教えて!goo」内の「恋愛」カテゴリに蓄積されている質問が含まれている。

本データセットでは、2,824件の質問に対する、約10,032件の結論部、補足部のトリプル（以降、便宜上 q-c-s と記す）を含むコーパスデータを用意し、モデルの学習を実施した。本データセットにおいては、平均して、質問に対する回答文は、人間によって選択された平均3.5件の結論部と補足部とで構成され、質問文、結論部、および補足部の平均の長さは、それぞれ482文字、41文字、および46文字であった。質問文には9,779の単語トークンがあり、回答文には6,317のトークンがあり、重複していたトークン数は4,096であった。

表 1 パラメータ α を変更した際の結果.

α	教えて!goo			nfl6		
	0	1	2	0	1	2
ROUGE-L	0.251	0.299	0.211	0.330	0.402	0.295
BLEU-4	0.098	0.158	0.074	0.062	0.181	0.023

表 2 教えて!goo データセットにおける ROUGE-L と BLEU-4.

	Seq2seq	CLSTM	Trans	HRED	NAGMWA	NAGM
ROUGE-L	0.238	0.260	0.278	0.210	0.291	0.299
BLEU-4	0.092	0.121	0.087	0.042	0.147	0.158

ii. nfl6

また, Yahoo! Answers の nfl6 データセットも使用した. これは, 公開されているなかで, 最大規模の英文の Non-Factoid 型 QA タスクの CQA データセットである. 特に, 多様な領域における回答文が必要なため, 「社会科学」, 「社会&文化」, 「芸術&人文科学」の各カテゴリの質問文を使用し, 13,579 件の質問文に対する 114,955 件の回答文を用いた.

さらに, スラングや URL, 記号のみからなる質問, 回答などを取り除くため, いくつかのストップワードを含む質問, 回答を取り除いた. このような質問, 回答を, 学習に含めると回答が生成されるときにノイズになることが多いためである. その上で, クラウドワーカを用い, 回答に対し, 10,299 件の結論部と補足部の文のペアに注釈を付けた.

そして, 2 値分類モデルを応用した回答文分類モデルを用いて, 文を結論クラスと補足クラスに 2 分類した. 具体的には, 最初に, 「The reason is」や「Because」などのフレーズで始まる文を, 補足部に分類するというルールを適用した. 次に, max-pooling 付きの BiLSTM を, 残りの未分類の文に対して適用した. つまり, 回答文を $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{N_a}\}$ とし, その特徴ベクトル $\mathbf{O}^a$ を生成した. その後, ロジスティックモイド関数を使用して, 結論文と補足文という 2 つの離散クラスへと写像される確率を出力した. この写像は, 上記の 10,299 個のラベル付き文を使用して分類エラーを最小限に抑えることによって学習した. その結果, 回答全体から 70,000 の質問-結論-補足トリプル (q-c-s) を自動的に取得した. なお, 質問数は, 11,768 件であった. 結果とし, 平均して約 6 つの結論と補足が 1 つの質問に割り当てられた. 質問, 結論, および補足の平均の長さは, それぞれ 46, 87, および 71 文字であった. 実験に先立ち, 分類器の性能を検証した. 人間の専門家が検証した結果, 約 81% は, 正確であると判断した (70,000 文のうち 56,762 文が, 正しいクラスに分類されていた). 質問には 15,690 語のトークンがあり, 回答には 124,099 個のトークンがあり, 両者のオーバーラップは 11,353 であった.

4.3 検証基準

教えて!goo データセットに対しては, 全データセットから, 9,532 件の q-c-s トリプルからなるトレーニング

表 3 nfl6 データセットにおける ROUGE-L と BLEU-4.

	Seq2seq	CLSTM	Trans	HRED	NAGMWA	NAGM
ROUGE-L	0.291	0.374	0.338	0.180	0.383	0.402
BLEU-4	0.081	0.141	0.122	0.055	0.157	0.181

表 4 教えて!goo データセットにおける定性評価.

CLSTM				NAGM			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
21	18	27	34	47	32	11	10

表 5 nfl6 データセットにおける定性評価.

CLSTM				NAGM			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
30	3	27	40	50	23	16	11

セットと 500 件の q-c-s トリプルからなるテストセットをランダムに 3 ペア準備した. そして, トレーニングセットを使用してモデルを学習した.

nfl6 データセットを使用した評価では, 前述したように, 回答文分類モデルを学習するために, 10,299 件の人間が注釈を付けたトリプルを使用し, ラベルのない回答文に分類器を適用した. そして 69,500 件の q-c-s トリプルからなるトレーニングセットと 500 件の q-c-s トリプルからなるテストセットを 3 ペア, ランダムに準備した. そして, トレーニングセットを使用してモデルを学習した.

学習後, テスト用のトリプルデータセット中の質問文を, 学習済みのモデルに入力し, 両方のデータセットの回答を生成し, 結果を正解文と比較した. 以下の結果は, 3 回の評価結果の平均値である.

なお, 両データセットには, 非常に多くの単語トークンが含まれているため, 生成時の softmax の計算に課題を持つ. この課題に対応するため, 多くの先行研究は, 単語の語彙を頻度に基づいて制限している [Yin 16, Yang 16, Vinyals 15] が, 語彙を縛ることは, 生成される回答の多様性を狭める. 一方, 本稿の目的とする Non-Factoid 型質問に適切に回答するには, 多様な回答が必要である. そのため本研究では, 語彙を制限せずに計算を高速化するために, 単語トークンの代わりにバイグラムトークンを使用した. 結果として, 教えて!goo データセットに 4,087 個のバイグラムトークンを用い, nfl6 データセットに 11,629 個のトークンを用いた. なお, ここでいうバイグラムトークンは文字グラムであり, 比較手法の中で transformer 以外についても適用している.

性能評価を行うために, 人間の定性的な評価以外に, 一般的な定量評価である単語単位での文一貫性を判定する ROUGE-L [Lin 04] と BLEU-4 [Papineni 02] を指標として使用した. これらは, Non-Factoid 型 QA を評価するためのパフォーマンスを測定するために使用される指標である [Song 17].

4.4 パラメータセットアップ

両データセットについて, さまざまなパラメータ値を試し, 結果としてバイグラムトークンのベクトルサイズ

を 500 に設定し、**C** および **S** のベクトルサイズを 500 に設定し、**CLSTM** モデルのトピック数は 15 にした。

また、式 (2) で α を 0 から 2.0 に変え、 L_s にあたえる影響を確認した。表 1 に結果を示す。表によると、 α がゼロの場合、結果は **Seq2seq** モデルの結果とほぼ同じであった。一方、 α の値を上げると、アンサンブルネットワークがより重視されるが、生成された結果の文法性も低下した。そのため、 α は、精度が最も高い 1.0 に設定した。この結果は、提案するアンサンブルネットワークが回答生成の精度に貢献していることを明確に示している。

4.5 実験結果

i. 精度

教えて!goo に対する結果を表 2 に示し、さらに、**nfL6** に対する結果を表 3 に示す。これらの結果は、**CLSTM** モデルが **Seq2seq** モデルよりも優れていることを示している。これは、トピックなどの文脈の特徴を踏まえて、質問文の文脈を踏まえた回答文を生成できるためである。**Trans** モデルもまた、**Seq2seq** モデルよりも良いモデルであることが確認された。これは、**Seq2seq** モデルよりも効果的に、質問文から結論部および補足部に対する文脈を学習し、用いることができるからである。**HRED** モデルは、十分な精度を示すことができなかった。この結果から、回答文を順次生成していくやり方では、最初のステートメント（質問）の元の意味に従う後続のステートメント（結論、さらに、補足）を生成するのが難しいことを示している。

NAGMWA モデルは、**NAGM** モデルを除く他の方法よりも精度が非常に良い。これは、結論部と補足部、およびそれらの組合せが、質問文に一致する回答文を生成するからである。つまり、回答文中の結論部と補足部は互いに一貫しており、Non-Factoid 型質問に割り当てられた複数の多様な結論部-補足部の組合せの存在による回答生成時の混乱を回避することができていることが分かる。

最後に、**NAGM** は、従来のアテンション機構を持つ比較手法や **NAGMWA** モデルよりも一貫して優れている。その **ROUGE-L** および **BLEU-4** スコアは、**CLSTM** のスコアよりもはるかに高くなっている。これは、**NAGM** は、質問と結論部または補足部の近さ、および結論部と補足部の組合せの近さに加え、結論から補足までの文脈を評価し、より流暢な文を生成できるからである。

ii. 人間による評価

クラウドソーシングされた評価者による [Li 16] らの検証を参考にし、**CLSTM** モデルの出力と **NAGM** モデルの出力の妥当性を判断するために人間による評価を実施した。[Li 16] の実験との違いとして、我々は、教えて!goo の QA コミュニティサービスに経験のある専門家を雇い、評価をした。彼らは、QA コミュニティに投稿される回答について詳しいため、結果の妥当性を判断するに適切である。評価において、各専門家には、評価対象システ

ムについて伝えず、50 の質問を評価させ、結果は、各専門家のスコアを平均した。

専門家は、トレーニングセットに含まれていない質問を AI システムに質問し、回答を評価した。なお、質問ごとに 1 つの回答がシステムから出力される。評価は **Document Understanding Conference (DUC)** で用いられている評価基準^{*3}を参考に、以下の 4 基準を用いた。(1) 回答の内容が質問と一致し、文法に問題がない。(2) 内容は適切であるが、文法が不十分である。(3) 内容は適切ではないが、文法には問題ない。(4) 内容も文法も不適切である。

表 4 と表 5 に結果が示されている。数値はパーセンテージ表示である。表 6 には、質問文と回答文の例が示されている。なお、教えて!goo の質問文は非常に長く、質問事項の前に長い背景説明文が含まれているため、要約したものを掲載している。

これらの結果から、人間の専門評価者が **CLSTM** の出力結果よりも、**NAGM** からの出力結果に強く満足していることが分かる。これは、表 6 に示されているように、**NAGM** が **CLSTM** よりも長くて優れた質問に沿った文を生成したためである。**NAGM** は文法的に良い回答文を生成し、その結論部と補足部は質問文と良く一致し、補足部は結論部に沿ったものを生成している。

なお、**NAGM** が生成した回答長は教えて!goo データセットで平均 61.9 文字、**nfL6** データセットで平均 112.2 文字であった。一方、**CLSM** が生成した回答長は教えて!goo データセットで平均 50.4 文字、**nfL6** データセットで平均 103.0 文字であった。

iii. コーパス中に含まれない回答文の生成

エンコーダ・デコーダモデルは、学習用コーパスにある回答文を再生成する傾向がある。一方、**NAGM** は、文脈と文の組合せを考慮したアンサンブルネットワークにより、コーパスに存在しない回答を生成し得る。

表 6 にいくつかの例を示す。たとえば、**NAGM** によって生成された回答#1 は、学習用コーパスに含まれていない。

この回答は、コーパスにある次の 3 つの文の太字の部分から生成されたと考えられる：(1)「あなたの側からは何もしないほうが良いと思います。彼からの反応がなければ、次に会う機会があっても何もしないほうがいい。」(2)「彼にアプローチしてみるのも良いかもしれないですね。あまり悲観的な想像ばかりせず、前向きに考えてみてはいかがでしょう。」(3)「普段考えているけれども、口に出して言わないことを、恋人に伝えてみませんか?… ケンカをした時こそ好きな気持ちや相手を大切に思う気持ちを伝えることが大事なのだと思いますよ。」

コーパス中に含まれていない回答文の生成については、推論時に新規投稿された質問の持つ文脈と、学習コーパ

*3 <http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt>

表 6 CLSTM モデルおよび NAGM モデルによって生成された回答文の例。質問 ID「1」は教えて!goo データに適用した結果であり、質問 ID「2」は nll6 データに適用した結果である。

ID	質問	CLSTM モデルによって生成された回答	NAGM モデルによって生成された回答
1	多くの人が、喧嘩は恋愛の別れの一例だと言う。最近、私たちはしばしばお互いに喧嘩する。しかし、数分以内に私たちは和解決し、私たちが口論したことについてささやう。そのような「頻繁な喧嘩」は私たちの間の関係を維持するのに良いのだろうか?それとも、別れを告げないように、お互いに喧嘩するのをやめるべきか?	相手に気持ちを伝えることを、それとなく聞いてみてはいかがでしょうか。相手にも、あなたのことを知ってもらえるようになるかもしれません。	何もしないほうが良いと思います。あまり悲観的な想像ばかりせずに、前向きに考えてみてはいかがでしょうか。ケンカをした時こそ好きな気持ちや相手を大切に思う気持ちを伝えることが大事なのだと思いますよ。
2	How I can find my love?	Feed him lots of fresh milk and change his diaper often. God hears prayers that are sincere.	Well, in my opinion, u can't get a lover without any effort...u gotta do smth! Otherwise don't care for the results, just go ahead and do it.

ス内の質問文の文脈のわずかな違いに対処できる可能性があり、Non-Factoid 型 QA システムにとって重要であるため、本結果は価値を持つと考える。

5. 結 論

本稿では、自然言語処理の重要タスクである、Non-Factoid 型質問に対する結論・補足の組合せ回答生成に取り組んだ。

特に、結論デコーダの出力系列の文脈を補足デコーダの出力系列に反映するアーキテクチャであるアンサンブル・ネットワークを提案した。アンサンブル・ネットワークは、質問文のエンコーダの出力系列と各デコーダの出力系列の近さと、両方のデコーダの出力系列の組合せも評価する。検証の結果、このタスクでは、提案手法が従来のエンコーダ・デコーダモデルに基づく手法よりも一貫して優れていることがわかった。

現在、提案手法を一般化して、結論・補足の組合せ回答以外のはるかに長い文章生成を目指している。たとえば、5 行の単語列で構成される古典的な日本の詩のジャンルである「短歌」の生成に適用できると考えており、適用検討を進めている。

◇ 参 考 文 献 ◇

- [中辻 19] 中辻 真, 奥井 颯平, 藤田 明久: LSTM を用いた Non-Factoid 型長文回答構築手法, 電子情報通信学会論文誌, Vol. J102-D, No. 4, pp. 267–276 (2019)
- [Bahdanau 14] Bahdanau, D., Cho, K., and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *CoRR*, Vol. abs/1409.0473, (2014)
- [Ennis 91] Ennis, R.: Critical Thinking: A Streamlined Conception, in *Teaching Philosophy*, pp. 5–25 (1991)
- [Ghosh 16] Ghosh, S., Vinyals, O., Strophe, B., Roy, S., Dean, T., and Heck, L.: Contextual LSTM (CLSTM) Models for Large Scale NLP Tasks, *CoRR*, Vol. abs/1602.06291, (2016)
- [Jia 17] Jia, R. and Liang, P.: Adversarial Examples for Evaluating Reading Comprehension Systems, in *Proc. EMNLP'17*, pp. 2021–2031 (2017)
- [Joshi 17] Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L.: TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, *CoRR*, Vol. abs/1705.03551, (2017)
- [Li 16] Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J.: Deep Reinforcement Learning for Dialogue Generation, in

- Proc. EMNLP'16*, pp. 1192–1202 (2016)
- [Lin 04] Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries, in *Text Summarization Branches Out: In: Proc. ACL-04 Workshop*, pp. 74–81 (2004)
- [Liu 18] Liu, X., Duh, K., and Gao, J.: Stochastic Answer Networks for Natural Language Inference, *CoRR*, Vol. abs/1804.07888, (2018)
- [Nakatsuji 20] Nakatsuji, M. and Okui, S.: Answer Generation through Unified Memories over Multiple Passages, in *Proc. IJCAI'20*, pp. 3823–3829 (2020)
- [Nakatsuji 20b] Nakatsuji, M. and Okui, S.: Conclusion-Supplement Answer Generation for Non-Factoid Questions, in *Proc. AAAI'20* (2020)
- [Nguyen 16] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L.: MS MARCO: A Human Generated MACHINE Reading COMprehension Dataset, in *Proc. Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 Co-located with NIPS 2016* (2016)
- [Nogueira 19] Nogueira, R., Yang, W., Lin, J., and Cho, K.: Document Expansion by Query Prediction, *CoRR*, Vol. abs/1904.08375, (2019)
- [Papineni 02] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J.: BLEU: A Method for Automatic Evaluation of Machine Translation, in *Proc. ACL'02*, pp. 311–318 (2002)
- [Qiu 15] Qiu, X. and Huang, X.: Convolutional Neural Tensor Network Architecture for Community-based Question Answering, in *Proc. IJCAI'15*, pp. 1305–1311 (2015)
- [Rajpurkar 16] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P.: SQuAD: 100, 000+ Questions for Machine Comprehension of Text, *CoRR*, Vol. abs/1606.05250, (2016)
- [Rinott 15] Rinott, R., Dankin, L., Perez, C. A., Khapra, M. M., Aharoni, E., and Slonim, N.: Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection, in *Proc. EMNLP'15*, pp. 440–450 (2015)
- [Santos 15] Santos, dos C., Barbosa, L., Bogdanova, D., and Zadrozny, B.: Learning Hybrid Representations to Retrieve Semantically Equivalent Questions, in *Proc. ACL-IJCNLP'15*, pp. 694–699 (2015)
- [Serban 16] Serban, I. V., Sordani, A., Bengio, Y., Courville, A. C., and Pineau, J.: Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models, in *Proc. AAAI'16*, pp. 3776–3784 (2016)
- [Song 17] Song, H., Ren, Z., Liang, S., Li, P., Ma, J., and Rijke, de M.: Summarizing Answers in Non-Factoid Community Question-Answering, in *Proc. WSDM '17*, pp. 405–414 (2017)
- [Sutskever 14] Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, in *Proc. NIPS'14*, pp. 3104–3112 (2014)
- [Tan 16] Tan, M., Santos, dos C. N., Xiang, B., and Zhou, B.: Improved Representation Learning for Question Answer Matching, in *Proc. ACL'16*, pp. 464–473 (2016)
- [Tan 17] Tan, C., Wei, F., Yang, N., Lv, W., and Zhou, M.: S-Net: From Answer Extraction to Answer Generation for Machine Reading Comprehension, *CoRR*, Vol. abs/1706.04815, (2017)
- [Vaswani 17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J.,

- Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention Is All You Need, in *Proc. NIPS'17* (2017)
- [Vinyals 15] Vinyals, O. and Le, Q. V.: A Neural Conversational Model, *CoRR*, Vol. abs/1506.05869, (2015)
- [Wang 07] Wang, M., Smith, N. A., and Mitamura, T.: What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA, in *Proc. EMNLP-CoNLL'07*, pp. 22–32 (2007)
- [Wang 18] Wang, Y., Liu, K., Liu, J., He, W., Lyu, Y., Wu, H., Li, S., and Wang, H.: Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification, in *Proc. ACL'18*, pp. 1918–1927 (2018)
- [Yang 15] Yang, Y., Yih, W., and Meek, C.: WikiQA: A Challenge Dataset for Open-Domain Question Answering, in *Proc. EMNLP'15*, pp. 2013–2018 (2015)
- [Yang 16] Yang, Z., Yuan, Y., Wu, Y., Cohen, W. W., and Salakhutdinov, R.: Review Networks for Caption Generation, in *Proc. NIPS'16*, pp. 2361–2369 (2016)
- [Yin 16] Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., and Li, X.: Neural Generative Question Answering, in *Proc. IJCAI'16*, pp. 2972–2978 (2016)
- [Yu 14] Yu, L., Hermann, K. M., Blunsom, P., and Pulman, S.: Deep Learning for Answer Sentence Selection, *CoRR*, Vol. abs/1412.1632, (2014)
- [Yu 18] Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V.: QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension, in *Proc. ICLR'18* (2018)

〔担当委員：坂地 泰紀〕

2021 年 6 月 13 日 受理

—— 著 者 紹 介 ——

中辻 真(正会員)

2001 年京都大学工学部数理工学卒。2003 年京都大学大学院情報学研究科システム科学専攻博士前期課程修了。2003 年 NTT 入社。2010 年京都大学大学院情報学研究科社会情報学専攻博士課程修了。2013 年度米国レンセラー工科大学研究員。2015 年より NTT レゾナント株式会社。2015 年度人工知能学会論文賞などを受賞。現在、深層学習を基にした対話システムおよび AI 基盤の研究開発、事業創造に従事。電子情報通信学会等の各会員、博士（情報学）。

八島 浩文(正会員)

2004 年慶應義塾大学総合政策学部総合政策学科卒業。2006 年上智大学大学院外国語学研究科国際関係論専攻博士前期課程修了。2006 年以降、商社（官庁営業担当）および外資系消費財企業（政府渉外担当）等を経て、2016 年 7 月 NTT レゾナント株式会社入社（AI 担当）。2016 年以降、自然言語処理技術の技術動向調査等に従事。