質問の意図を特定するニューラル質問生成モデル

大塚　淳史† 　西田　京介† 　斉藤いつみ† 　浅野　久子† 　富田　準二†

† 日本電信電話株式会社　NTT メディアインテリジェンス研究所　〒 239–0847 神奈川県横須賀市光の丘 1-1

E-mail: †{otsuka.atsushi,nishida.kyosuke,saito.itsumi,asano.hisako,tomita.junji}@lab.ntt.co.jp

あらまし　質問応答システムでは，ユーザの質問意図が曖昧である場合，回答となる情報を特定できず，システムの回答精度が低下するという課題があった．本論文では，質問応答システムにおいてユーザの質問意図が曖昧な場合に，質問対象となる文書の内容を基に明確な質問を生成する改訂質問生成（RQG) タスクを新たに定義する．そして，本問題に適用可能なニューラル質問生成モデルを新たに提案する．提案モデルでは，質問応答技術で用いられる機械読解と自然文生成のニューラルネットワークを組み合わせることにより，文書の中から質問に関連する情報を抽出し，抽出した情報から具体化された改訂質問（RQ）を生成する．提示された具体的な質問を閲覧することで，ユーザは自らの質問意図を明確化できると同時に，質問応答システムから所望の回答を得られる質問を入力できるようになる．日本語の機械読解データを基にモデルの学習・評価を行い，ベースライン手法に比べて質問を精度良く改訂出来ることを確認した．

キーワード　質問応答，質問生成，機械読解，ニューラルネットワーク

1. はじめに

近年，スマートフォンやスマートスピーカー等のデバイス上で，ユーザが自然言語によって入力した質問に対する回答をコンピュータが自動で行う質問応答技術が注目を集めている．質問応答は，ユーザの質問の意図を自動で理解し，データベースやテキストといった知識源から回答となる情報を探し出すことで実現される．特に，自然言語で入力された質問に対して，同じく自然言語で記述された部分文書内から回答となる部分を直接抽出する機械読解型の質問応答 [8] では，ディープラーニングを用いることで人に匹敵する回答精度を達成出来ることが報告されている [18, 24]．

質問応答において，ユーザの質問に正しく回答するためには，質問の意図を正確に理解する必要がある．しかし，その前提として，ユーザが入力する質問には，回答のために必要な情報が不足無く含まれていなくてはならない．入力された質問が曖昧であったり，短すぎる場合，回答が一意に決定できないため回答の精度が低下するという問題がある．コールセンター等の人同士の質問・回答のやり取りを例にとると，質問者が曖昧な質問や要求を行う場面がこの問題に相当する．この問題の解決にあたり，我々はコールセンタのオペレータが行う「自身が持つマニュアル等のテキスト情報と質問の内容を照らし合わせ，質問意図を推測した上で確認を行う」プロセスに着目した．例えば，「解約料金はいくら」という質問に対して，オペレータは「プラン A の解約料金でしょうか」のように，質問者の質問を具体化した候補を提示することで，質問意図を明確化している．質問応答システムにおいても，ユーザの質問が曖昧な場合，与えられた情報のみで回答するのではなく，ユーザの質問をより具体化した質問を例示することで，質問の情報不足や曖昧性を解消させることによる回答精度の向上が期待できる．

本論文では，質問応答システムにおいて，ユーザが自然文で入力した質問に対して，内容を具体化した質問を生成する技術を提案する．ここで，システムが具体化した質問を**改訂質問（Revised Question : RQ）**と呼び，改訂質問を生成するためのタスクを**改訂質問生成（Revised Question Generation : RQG）**と定義する．RQG の流れを図 1 に示す．RQG モデルに入力質問と質問対象のテキスト（パッセージ）を入力する．RQG モデルは入力質問とパッセージの内容を比較し，関連する情報を抽出する．その情報を注視して RQ を生成する．

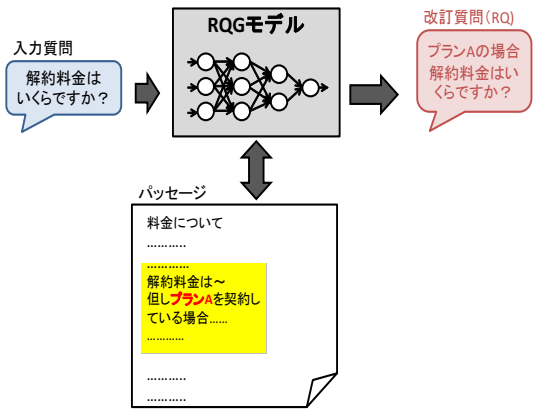


図 1: 短い入力質問とパッセージの内容から，入力質問を具体化した質問（RQ) を作成する改訂質問生成（RQG) タスク

質問応答システムのユーザは，質問意図が曖昧なときでも，RQG モデルが提示する RQ によって質問意図を明確化することができる．自身の質問意図が RQ と同じであった場合，提示された RQ をそのまま再質問することで，所望の回答が得られる．本論文では，評価実験によって，日本語の機械読解用のデータセットを使用し，短い質問を入力したとき，RQG モデ

ルにより，内容が具体化された RQ が生成できることを示す．

本論文の貢献：本論文は，質問応答システムにおいて，以下の貢献を果たした．

- 新しいタスク**改訂質問生成（RQG）**を定義した．RQG では，入力質問に対して，質問対象の文書内容に基いて，質問意図を具体化した改訂質問の例を生成し，ユーザに提示する．
- RQG を実施可能なニューラルネットワークモデルを提案した．提案モデルは言語生成と機械読解のニューラルネットワークを組み合わせることで，End-to-End でのモデル学習を可能とする．
- 日本語の機械読解データを基に提案モデルの学習・評価を行い，ベースライン手法に比べて質問を精度良く改訂出来ることを示した．

本論文の構成は以下の通りである．まず，2. で，本論文上での用語定義や問題設定について明らかにする．次に 3. で提案モデルについて説明し，その学習方法については，4. で詳述する．5. で評価実験結果を示す．6. では，関連研究について述べ，最後に 7. で本論文のまとめを行う．

2. 問 題 定 義

本節では，本論文が取り組むタスクまたそれに関連する用語について定義を行う．

[定義 1] 入力質問は，自然言語で記述された文である．質問文は形態素解析等の処理により，$Q = \{q_0, q_1, ..., q_{J-1}\}$ といった単語トークン集合に分割できる．

[定義 2] パッセージは，自然言語で記述された文である．パッセージは数百語程度の単語から構成され，入力質問と同様に $X = \{x_0, x_1, ..., x_{T-1}\}$ といった単語トークンの集合で表わされる．ここで，パッセージは，入力質問の回答となる情報を含むものとする．

[定義 3] 改訂質問（Revised Question : RQ）は，入力質問の内容が具体化された文 $RQ = \{y_0, y_1, ..., y_{K-1}\}$ である．

[定義 4] 改訂質問生成（Revised Question Generation : RQG）は，入力質問とパッセージを入力とし，RQ を出力するタスクである．RQ は入力質問の内容を具体化し，かつパッセージ内容に沿った内容となっている．

[定義 5] RQG モデルは，入力質問の単語系列 $Q \in V$ と，パッセージの単語系列 $X \in V$ を入力とし，RQ の単語系列 $RQ \in V$ を出力するニューラルネットワークモデルである．ここで，V は，RQG モデルが扱う語彙空間である．

3. RQG モデル

本節では，RQ を生成するための，RQG モデルについて説明する．RQG モデルの構成を図 2 に示す．RQG モデルは，Encode Layer，Matching Layer，Decode Layer の 3 つの階層から構成される．Encode Layer と Decode Layer は言語生成のモデルである Seq2Seq [21, 23]，Matching Layer は，機械読解タスクで用いられる BiDAF [18] をベースとしている．以降は，RQG モデルの各層について詳述する．

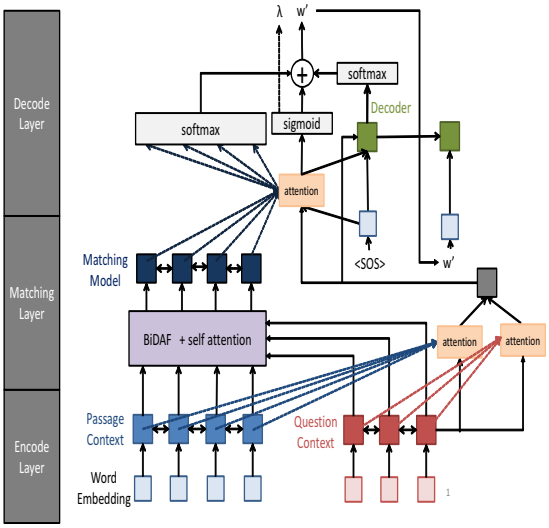


図 2: RQG モデル

3.1 Encode Layer

RQG モデルの Encode Layer では，パッセージと入力質問を Recurrent Neural Network （RNN）によってエンコードする処理を行う．パッセージと入力質問はそれぞれ d 次元の単語ベクトルの系列，$x = \{e_{x_0}, e_{x_1}, .., e_{x_{T-1}}\}$，$q = \{e_{q_0}, e_{q_1}, ..., e_{q_{J-1}}\}$ で表されているものとする．単語ベクトルは Glove [15] や FastText [1] といった単語ベクトル生成モデルによって事前に作成する．単語ベクトルは RQG モデルの学習データへフィッテングさせることを目的に，Bi-LSTM でエンコードする前に 2 層の Highway Newwork [20] によって，$\mathbb{R}^d$ に写像する．

パッセージ，入力質問をそれぞれ RNN に入力し，パッセージ，入力質問のコンテキスト行列 $H \in \mathbb{R}^{2d \times T}$，$U \in \mathbb{R}^{2d \times J}$ を得る．RNN には，隠れ状態の長期記憶を可能とする Long Short Term Memory (LSTM) [9] を用いる．ここで，最初に入力した単語ベクトルの影響が小さくなることを防ぐために，先頭から順に LSTM に入力するのと同時に，最後尾から逆順に LSTM に入力した状態を併用する BiDirectional LSTM (Bi-LSTM）[17] を使用する．なお，パッセージと入力質問をエンコードする Bi-LSTM のパラメータは共通である．

3.2 Matching Layer

Matching Layer では，Encode Layer でエンコードしたパッセージと入力質問との照合し，パッセージ中から，入力質問に関連する領域を発見する．本論文では，3.3 節で説明するように，改訂質問を生成する Decode layer において，パッセージ中に含まれる単語を抽出するための CopyNet の概念 [3,7] を取り入れる．そこで Matching Layer では，デコーダおよび CopyNet への入力として，質問とパッセージのマッチングを 2 種類行って出力する．以降はそれぞれのマッチングについて説明する．

3.2.1 デコーダの初期状態を作成するマッチング

Seq2Seq では，エンコーダの隠れ状態をデコーダの RNN の隠れ状態の初期値に引き継ぐという特徴がある．RQG モデル

は，入力質問をベースに，パッセージの情報を参照して，入力質問を詳細化する文を生成するモデルである．そのため，入力質問のエンコード状態をベースとしてデコーダにより生成することが望ましい．本論文では，入力質問のコンテキスト行列 U の系列最後のベクトル U_{J-1} が入力質問のエンコード状態を表したベクトルになる．しかしながら，入力質問のエンコード状態だけでは，パッセージの内容は全く考慮できない．そこで，アテンション機構 [12] を用いて，パッセージの情報も考慮した隠れ状態のベクトルを作成する．

入力質問のエンコーダの隠れ状態 U_{J-1} と，パッセージのコンテキスト行列 H に対して，パッセージとのアテンションを考慮したベクトル $\hat{H}_U \in \mathbb{R}^{2d}$ は以下の通り計算できる．

$$\alpha_t = \mathrm{softmax}_t(H_t \cdot U_{J-1}) \tag{1}$$

$$\hat{H}_U = \sum_t \alpha_t H_t \tag{2}$$

ここで，$H_t \in \mathbb{R}^{2d}$ は，パッセージのコンテキスト行列 H の各系列での隠れ状態であるベクトルを表している．

同様に，入力質問のコンテキストベクトル U_{J-1} と入力質問のコンテキスト行列 U とのアテンションベクトル $\hat{U}_U \in \mathbb{R}^{2d}$ を作成する．これは入力質問のコンテキスト自身でアテンションを取ることになり，入力質問中の重要な単語を考慮するためのものである．

２つのアテンションベクトルル $\hat{H}_U$，$\hat{U}_U$ からデコーダの初期状態とする隠れ状態ベクトル h_{d0} は以下の式で計算する．

$$h_{d0} = f(W_m[\hat{H}_U; \hat{U}_U] + b_m) \tag{3}$$

ここで，$W_m \in \mathbb{R}^{2d \times 4d}$，$b_m \in \mathbb{R}^{2d}$ は学習パラメータである．また活性関数 f は Leaky ReLU を使用する．

3.2.2 BiDAF によるマッチングモデル構築

BiDAF は，ニューラルネットワーク上で，２文間の関連する領域を発見するためのモデルであり，機械読解タスクでトップクラスの性能を達成しているモデルの一つである．本論文では，BiDAF によって構築したパッセージと入力をマッチングさせた層（Modeling 層）を Decode Layer でコンテキストベクトルとのアテンションに用いる．

まず，エンコーダの出力である系列長 T のパッセージのコンテキスト行列 $H \in \mathbb{R}^{2d \times T}$ と，系列長 J の入力質問のコンテキスト行列 $U \in \mathbb{R}^{2d \times J}$ を BiDAF の Attetntion 層へ入力する．Attention 層ではまず，パッセージと入力質問の単語の類似度行列を計算する．パッセージの t 番目の単語と，入力質問の j 番目の単語の類似度を

$$S_{tj} = w_s^T[H_t; U_j; H_t \odot U_j] \tag{4}$$

と定義する．ここで $w_s^T \in \mathbb{R}^{6d}$ は学習パラメータである．$[;]$ は行列の水平方向への連結を表しており，$\odot$ は要素積である．S_{tj} を要素に持つ類似度行列 $S \in \mathbb{R}^{T \times J}$ を作成する．類似度行列から，パッセージから入力質問へのアテンションと入力質問からパッセージへのアテンションの２方向のアテンションを計算する．

パッセージから入力質問へのアテンションでは，パッセージ中の単語について，入力質問の単語で重み付けしたベクトルを計算する．t 番目のパッセージ単語に対応するアテンションベクトル $\check{U}_t \in \mathbb{R}^{2d}$ は，

$$a_{tj} = \mathrm{softmax}_j(S_t) \tag{5}$$

$$\check{U}_t = \sum_j a_{tj} U_j \tag{6}$$

となる．質問からパッセージへのアテンションでは入力質問のいずれかの単語に強く関連する単語の重みを付けたベクトル $\check{h}$ をパッセージの系列長 T 分並べた行列 $\check{H} \in \mathbb{R}^{2d \times T}$ を作成する．ベクトル $\check{h}$ は以下の通り計算する．

$$b = \mathrm{softmax}_t(\mathrm{max}_j(S)) \tag{7}$$

$$\check{h} = \sum_t b_t H_t \tag{8}$$

BiDAF の Modeling 層では，Attention 層で計算したアテンション行列 $G \in \mathbb{R}^{8d \times T}$ を入力とする Bi-LSTM によって BiDAF の出力 $M \in \mathbb{R}^{2d \times T}$ を得る．本論文では，アテンション行列 G は，パッセージのコンテキストベクトル H_{T-1} とパッセージのコンテキスト行列 H との self-attention [24] をとったアテンションベクトル $\hat{H}_H \in \mathbb{R}^{2d}$ を加えた次式で表わされるものを使用する．

$$G = [H; \check{U}; H \odot \check{U}; H \odot \check{H}; \hat{H}_H] \in \mathbb{R}^{10d \times T} \tag{9}$$

3.3 Decode Layer

Decode Layer では，Matching Layer で作成したマッチング結果から RQ を作成するための単語系列を出力する．デコードにはアテンション機構付きの LSTM と sotmax 関数を用いる．デコーダの LSTM の隠れ状態の初期値は式 3 で計算した h_{d0} を用いる．入力は，Encode Layer と同様の単語埋め込みベクトルとする．デコーダにはまず文の始端を表すトークン ($< BOS >$) の単語埋め込みベクトルを入力する．その後は生成した単語 y_s の埋め込みベクトル e_{y_s} を与える．

デコーダの LSTM に与える s 番目の入力 $\hat{z}_s \in \mathbb{R}^{3d}$ は，1 つ前に生成した単語の埋め込みベクトル $e_{y_{s-1}} \in \mathbb{R}^d$ と LSTM の隠れ状態 $h_{d(s-1)} \in \mathbb{R}^{2d}$ の合成ベクトルと，BiDAF のマッチングモデル $M \in \mathbb{R}^{2d \times T}$ でアテンションをとった次式で得る．

$$\hat{h}_s = f(W_d[e_{y_{s-1}}; h_{d(s-1)}] + b_d) \tag{10}$$

$$\alpha_{st} = \mathrm{softmax}_t(M_t \cdot \hat{h}_s) \tag{11}$$

$$\hat{c}_t = \sum_s \alpha_{st} M_t \tag{12}$$

$$\hat{z}_s = [e_{y_{s-1}}; \hat{c}_s] \tag{13}$$

ここで，$W_d \in \mathbb{R}^{2d \times 3d}$，$b_d \in \mathbb{R}^{2d}$ は学習パラメータ，f は活性関数である．$M_t \in \mathbb{R}^{2d}$ は，マッチングモデルの各系列を表すベクトルである．デコーダが使用する LSTM の隠れ状態 h_{ds} は，次式のように更新する．

$$h_{ds} \leftarrow \mathrm{LSTM}(h_{d(s-1)}, \hat{z}_s) \tag{14}$$

デコーダの LSTM と softmax 関数により，出力単語に関する

生成確率分布 $P_G(y_s|y_{<s}, X, Q)$ を得る．

Seq2Seq では，生成確率が最大の要素に対応する単語を出力するが，RQG の場合，なるべくパッセージ内に含まれる単語を出力するようにするために，CopyNet [3,7] を応用した手法を導入する．CopyNet とは，単語の生成確率を LSTM の出力の外からも与えることで，エンコード側の単語をそのまま生成しやすくするニューラルネットワークモデルであり，自動要約や対話処理などで高い効果が挙げられることが報告されている．本論文では，アテンションベクトル $\hat{c}_s$ を計算する際の単語重み α_{st} を使用した単語の生成確率による Copy 機能を導入する．Copy による単語の生成確率 $P_c(y_s|y_{<s}, X, Q)$ は次式より得る．

$$P_c(y_s|y_{<s}, X, Q) = \sum_t \mathbb{1}(y_s = X_t)\alpha_{st} \tag{15}$$

ここで，$\mathbb{1}(y_s = X_t)$ は，生成する単語 y_s がパッセージの t 番目の単語 X_t と一致する時は 1，それ以外のときは 0 を返す関数である．

最終的な単語の生成確率 $P(y_s|y_{<s}, X, Q)$ は重み λ_s による以下の加重平均によって計算する．

$$\begin{aligned} &P(y_s|y_{<s}, X, Q) \\ &\quad = \lambda_s P_c(y_s|y_{<s}, X, Q) + (1-\lambda_s)P_G(y_s|y_{<s}, X, Q) \end{aligned} \tag{16}$$

重み λ_s は，アテンションベクトル $\hat{c}_s$ から次式によって計算する．

$$\lambda_s = \sigma(W_\lambda \hat{c}_s + b_\lambda) \tag{17}$$

ここで，$W_\lambda \in \mathbb{R}^{1\times 2d}$，$b_\lambda \in \mathbb{R}^1$ は学習パラメータ，σ はシグモイド関数を示す．

4. RQG モデル学習

本節では，3. で説明した RQG モデルについて，モデルの学習方法について説明する．まず，RQG モデルを学習するためのマルチタスク学習について説明し，次に，学習データ作成について述べる．

4.1 マルチタスク学習

RQG モデルでは，出力単語の生成確率 P を教師データに近づけるパラメータ θ を学習する．ここで，式 16 の通り，出力単語の生成確率を求めるためには適切な λ_s が設定されている必要がある．本論文では，適切な λ_s が自動で算出されるように，出力単語の生成確率と同時に λ_s も同時に学習するマルチタスク学習を導入する．RQG モデルの学習では，単語生成についての損失 L_g と λ_s についての損失 L_λ の和 $L(\theta) = L_g + L_\lambda$ を最小化する．

重み λ_s は，1 に近い値を取るほど，パッセージ中の単語からコピーされる確率が上がることを示している．そのため学習時には，生成する単語がパッセージ内に含まれる単語であれば 1，それ以外は 0 のラベルを付与する．2 値のラベルを付与することで，λ_s は，$\hat{c}_s$ が生成する単語がパッセージ内に含まれる単語であるかを予測する確率となる．λ_s が 1 に近い値であるときは，生成されて欲しい単語がパッセージ内にある確率が高いと判断され，パッセージから生成した生成確率 P_c が強く考慮されるようになる．損失 L_λ は 2 値のクロスエントロピーによって次式の通り計算する．

$$L_\lambda = -\frac{1}{N}\sum_i (r_i \log p^r + (1-r_i)\log(1-p^r)) \tag{18}$$

ここで，N はミニバッチのサンプル数，r_i は i 番目のサンプルデータが正例のとき 1，負例のとき 0 をとる適合度である．

単語生成の損失 L_g は，負の対数尤度である次式によって計算する．

$$L_g = -\frac{1}{N}\sum_i r_i(\sum_s p(y_s|y_{<s}, X)) \tag{19}$$

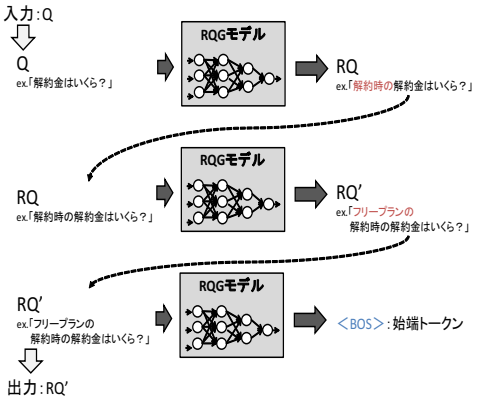


図 3: 部分生成による RQG の流れ

4.2 部分生成による RQG と学習データ作成

RQG では，入力の質問を具体化した RQ を生成する．入力質問は短く曖昧な質問を想定しているため，RQG モデルは，短い質問から長くかつ具体的な RQ となる文を生成する必要がある．本論文では，短い入力でも十分に長い文を生成するための手法として，**部分生成による RQG** を提案する．部分生成による RQG の流れを図 3 に示す．入力 Q に対して，RQG モデルは，Q を具体化した文 RQ を生成する．ここで，RQ は入力に対して，1 文節分だけ長くした文となるように学習する．次に，RQ を同じ RQG モデルに入力し，更に 1 文節分長くなった文 RQ' を得る．この操作を RQG モデルが始端トークン（$< BOS >$）を生成するまで繰り返し行う．始端トークンが生成されたとき，このときの入力 RQ' を RQG の最終的な生成文として出力する．なお，本論文では，部分生成を行わず，1 回の RQG モデルで十分に具体化された RQ を生成する手法を**全体生成**と呼ぶ．

RQG モデルを学習するためには入力質問と，入力質問を具体化した RQ を教師データとして作成する必要がある．しかしながら，質問文を作成するには人手のコストが大きい．本論文では，部分生成の特徴を生かして，具体的な質問文を用意すること無く RQG モデルを学習する方法を提案する．本手法では，機械読解などのコーパスに用意されている質問文を「具体化された質問文（RQ)」とする．まず，質問文を構文解析し，文節

表 1: データセットの平均件数およびトークン数

	件数		トークン数	
	パッセージ数	質問数	パッセージ長	質問長
train	4,000	66,073	181.4	21.9
test	500	8,247	176.2	21.8

表 2: 欠損文節数毎の評価データ件数

欠損文節数	1	2	3	4	5
データ件数	1,908	1,846	1,593	1,268	887
欠損文節数	6	7	8	9	10
データ件数	595	366	223	116	50

単位に分解する．そこから任意の 1 文節を取り出し入力質問とする，そして，入力文節に入力文節の 1 つ前の文節を結合したものを教師データとする．この操作によって，入力に対して 1 文節分長くなった文を生成するための学習データが 1 組作成される．この手順を文頭になるまで繰り返す．入力に文頭の文節を含まれる場合，教師データは始端トークンにすることで，RQ が十分な情報（文頭まで）を含んだ場合は始端トークンを生成して，生成を終了できるようにする．これを全文節のパターンで機械的に行うことによって，1 つの質問文から大量の入力と教師データのペアを作成することができる．

5. 評 価 実 験

本節では評価実験について述べる．まず，実験設定について説明し，次に実験結果について述べる．

5.1 データセット

評価実験では，我々が作成した日本語のニュース記事に関する，機械読解のデータセット（Jp-News）を用いる．Jp-News は機械読解のデータセットである SQuAD の形式に準拠しており，パッセージ，質問，回答から構成される．Jp-News は，広範囲なトピックに関するニュースをカバーしている．また，質問意図が明確で，回答がパッセージ中から一意に抽出できる質問によって構成されているという特徴がある．

本実験では，データセットの中のパッセージと質問を使用する．データセット Jp-News の平均件数，およびトークン数を表 1 に示す．データセットはモデル学習用（train）と評価用（test）に分かれている．

5.2 実験用モデル

実験用のモデルとして，Jp-News の train セットを用いて学習を行った RQG モデルを用いる．データセットの質問に対して，4.2 で説明した手法により，312,721 件のパッセージ，入力，出力の組からなる学習セットを作成した．

RQG モデルは，4 GPU により動作させ学習を行った．入力の単語埋め込みベクトルには Wikipedia のデータから，Glove によって作成した 100 次元のベクトルを用いる（語彙数:14,8631）．ニューラルネットワークの隠れ状態の層数は 100 に設定する．学習には Adam を使用し，ドロップアウト率は 0.2 に設定している．

5.3 評 価 指 標

評価には Jp-News の test セットを用いる．test セットの質問に対して，先頭 N 文節を削除した質問（欠損質問）を入力とし，文節を削除する前の元の質問を正解とする．RQG モデルに欠損質問を入力した時，元の質問に復元できれば入力に対して追加情報が付加された具体的な RQ が生成されたことになる．削除文節数 N は 1〜10 に設定する．このときのデータ数を表 2 に示す．なお，本実験で使用する質問は test セットの中で，句点を含まないかつ質問長 50 以下の質問に限定している．

入力した欠損質問が正しく復元できているかを評価する尺度としては BLEU [14]，Precision，Recall を用いる．BLEU は機械翻訳の自動評価を中心に用いられる評価尺度であり，N-gram の一致度にもとづいて，出力文と正解文の類似度を評価する．Precision は RQG モデルが出力した単語が，正解文に含まれる単語である割合である．Recall は RQG モデルが出力した文の単語集合の，正解文の単語集合に対する網羅度を示している．

5.4 比 較 手 法

提案手法 **(proposed)** は 3. 節および図 2 で示したモデルを 4.2 で説明した，入力を 1 文節づつ長くしていく（部分生成）学習によって作成した RQG モデルを用いる．

ベースラインとして，従来の Seq2Seq を用いて構築した RQG モデルを使用する．Seq2Seq モデルは，OpenNMT [10] のデフォルト設定であるアテンション機構付き･2 層の双方向 LSTM による Encoder-Decoder モデル [12] とした．

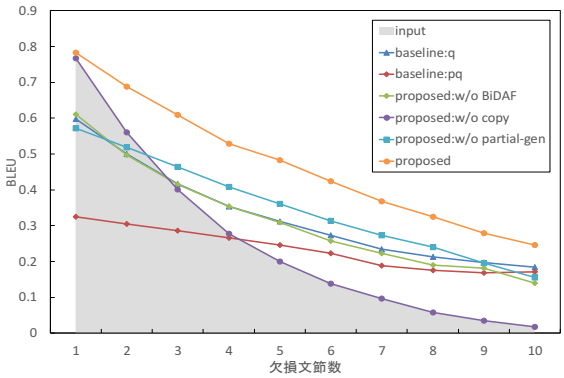
ここで，欠損質問のみを入力して，元の質問を生成するモデル **(baseline:q)** と，パッセージと欠損質問を文字列結合したものを入力とし，元の質問を生成するモデル **(baseline:pq)** の 2 種類のベースラインモデルを使用する．

比較手法は，提案手法に対してモデルと学習方法を変更したものを用いる．ニューラルネットワークの性能を評価するための比較手法として，Matching Layer で行っている BiDAF によるマッチングを行わず，パッセージのコンテキスト行列 H を Decode Layer のアテンションに用いたモデル **(proposed:w/o BiDAF)**，Decode Layer で CopyNet の機能を使用せず，LSTM の隠れ状態から出力した単語の生起確率のみで出力する単語を決定するモデル **(proposed:w/o copy)** を用いる．また，部分生成の有効性を評価するために，モデルは提案手法と同様とするが，学習時に部分生成ではなく，一度に元の質問を全て生成する（全体生成）するように学習を行ったモデル **(proposed:w/o partial-gen)** を使用する．

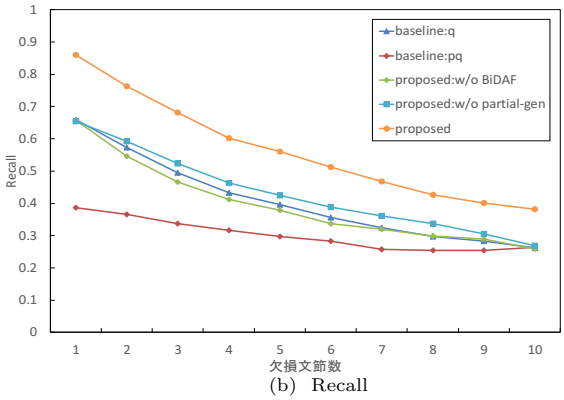
5.5 実 験 結 果

評価実験結果を図 4 に示す．横軸は欠損文節数であり，縦軸は文単位の BLEU, Recall，Precision のスコアを表している．ここで **input** は，入力である欠損質問と元の質問との BLEU スコアを示している．

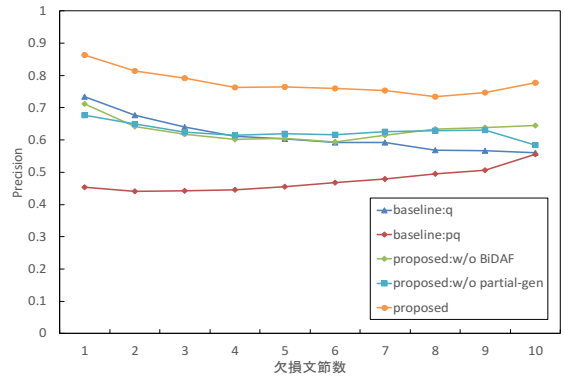
BLEU スコアについて，評価セットの全データを用いて提案手法 **(proposed)** と **input** および，ベースライン，比較手法との t-検定を実施したところ，いずれの手法に対しても，提案手法が有意にスコアが改善される（$p < 0.001$）ことが確認できた．



(a) BLEU



(b) Recall



(c) Precision

図 4: 欠損質問からの復元実験結果

[評価 1 欠損質問からの質問復元に関する評価] BLUE による評価実験では，BLEU スコアが **input** 以下のスコアとなっている場合，入力の欠損質問よりも ROG モデルの出力 RQ の方が元の質問と一致度が低くなるデグレードを起こしていることを意味している．欠損文節数 1 のとき，提案手法以外の手法ではデグレードを起こしている．欠損文節数が 1 の際には追加する情報は 1 文節分であり，その他は入力をそのまま出力するオートエンコーダーとしての役割が求められる．入力に欠損質問のみを用いた **baseline:q** は，パッセージ等の付加情報を使用していないため，Seq2Seq モデルが純粋なオートエンコーダーに近い性質を持つように学習が行われていると考えられる．

欠損文節数が多くなると，いずれの手法も BLEU スコアが低下していくが，それ以上に **input** の BLEU スコアが大きく低下していくことがわかる．これは，欠損質問が元の質問に対して非常に短い文になっていることを示している．欠損文節数が 5 を超えるといずれの手法でも **input** よりも高いスコアとなる．また，そこから欠損文節数が増えるほど **input** との差が大きくなる．このことから，RQG モデルは入力質問が極端に短い場合でも，パッセージの情報を利用することで質問を精度良く具体化できることが分かる．

[評価 2 Copy に関する評価] パッセージを用いていない **baseline:q** は，欠損文節数が多くなると Precision が低下していくが，パッセージを使用している他の手法については，Precision はほぼ一定または上昇している．これは，入力が短くなり新たな情報を追加した文を生成する際には，パッセージの情報を使用することが重要となることを示している．その中でも，パッセージ中の単語を Copy する機能を有するモデルはいずれの手法もベースラインよりも高い Precision となっている．

Copy を使用してない **proposed:w/o copy** では，BLEU スコアの結果が **input** と一致している．これは，**proposed:w/o copy** が評価セットの全ての場合において，入力をそのまま出力してしまっていることを示している．本論文で提案しているニューラルネットワークの学習方法では，生成の 1 単語目で文頭トークンが他の単語に比べて圧倒的に生成されやすくなる．Copy を導入した手法では，出力する単語はパッセージからコピーするかを重み λ によって制御しているため，デコーダの出力層が文頭トークンを生成しようとしても，λ によって無視される．一方，Copy を導入していない場合，出力層から単語がそのまま出力されるので，すぐに文頭トークンが出力されてしま，入力をそのまま出力するというモデルに学習されてしまっていると考えられる．

[評価 3 Matching Layer に関する評価] 比較手法 **(proposed:w/o BiDAF)** は，提案モデルと比較して Matching Layer で BiDAF を使用していない．そのため，出力層を計算する際のアテンションおよび，Copy する単語はパッセージとコンテキスト状態（LSTM の隠れ状態 h）のみ考慮される．実験結果では，パッセージと入力質問を考慮した BiDAF モデルを用いた提案手法が，**proposed:w/o BiDAF** よりも有意に良い結果となっている．このことから，RQG モデルにおいて，単語生成の際に，パッセージと入力質問のマッチング状態を考慮することは非常で重要であるといえる．

[評価 4 部分生成に関する評価] 提案手法 **(proposed)** と比較手法 **(proposed:w/o partial-gen)** は，RQG モデルのニューラルネットワークの構成は同じであるが，モデルの学習方法が異なる．**proposed:w/o partial-gen** は，1 回の RQG モデルへの入力で RQ を生成する全体生成であるのに対して，**proposed** は，1 回の RQG モデルへの入力では，入力に対して 1 文節分長い RQ を出力し，再度 RQ を RQG モデルへ入力する．これを始端トークンが生成するまで繰り返し行う部分生成によって RQ を生成する．

全体生成では，パッセージとの短い入力質問から 1 回のマッチングで長い RQ を生成する必要がある．一方で，部分生成の場合には，マッチングは入力の度に繰り返し行われ，またその

表 3: RQ を用いた機械読解例

passage			
気象庁によると、北陸地方は晴れや曇りとなっていますが、湿った空気や上空の気圧の谷の影響で、次第に大気の状態が不安定になる見込みです。このため１８日夕方から夜にかけて、局地的に雨雲が発達して雷雨となり、１時間に３０ミリ以上の激しい雨が降るおそれがあります。１９日朝までに降る雨の量は、多いところで８０ミリと予想されています。			
Q	雨が降ると予想されているのはいつまでか？	A	１９日の朝まで
RQ	１時間に３０ミリ以上の激しい雨が降ると予想されているのはいつまでか?	A	１８日の夕方から夜にかけて

度に入力質問が長くなってくためマッチング範囲も広くなる．この違いが生成の精度に影響していると考えられる．また，部分生成の場合は情報を追加する部分は 1 文節分であり，それ以降は入力をそのまま出力するオートエンコーダーの役割となることから，RQG のタスク自体が部分生成の場合簡易化されていることも影響していると考えられる．

[評価 5 機械読解への影響に関する評価] RQ を用いた機械読解の例を表 3 に示す．機械読解は BiDAF ベースのモデル [28] を使用している．天気予報に関するパッセージに対して，質問 (Q) を機械読解に入力した時の回答と，Q を RQG モデルに入力して得た RQ で機械読解を行ったときの回答を示している．

Q と RQ で機械読解の回答が変化していることがわかる．Q の「雨が降ると予想されているのはいつまでか？」に対して，RQ は「1 時間に 30 ミリ以上の激しい雨が降る」が条件として追加されている．機械読解はそれらの違いを正しく識別して，異なる回答を出力している．このことから，RQG モデルは入力質問に対して，パッセージ上で関連している部分を正しく認識し，より具体的な内容となる部分を追加するモデルになっていることがわかり，機械読解での質問を具体化するという本論文の目的は，RQ および RQG モデルによって達成できていることいえる．

6. 関 連 研 究

6.1 質問応答システムにおける質問に着目した研究

質問応答において質の良い質問は，精度の高い回答を得るためには必要不可欠である．質問応答での回答精度を高めるために，入力の質問の品質自体を改良するための研究が報告されている．Buck ら [2] らは，質問応答において，ユーザとブラックボックス化した質問応答システムの間に入るエージェントを設定し，ユーザの入力質問を質問応答システムが回答できる質問に変換して，回答を得る技術を提案している．Kumar ら [11] らは，対話型の視覚的質問応答システムにおいて，省略などで不完全な状態の質問が入力されたとき，これまでの対話のコンテキスト情報を参考に質問で不足している情報を補完することで回答精度が改善されることを報告している．

近年，SQuAD [16] などの機械読解のオープンデータにおいて，機械読解の逆問題として，パッセージとパッセージ内の正解となる区間を与えて，質問文を自動的に生成する試みが行われている．Du ら [5] は，SQuAD データを対象に，アテンション付き Seq2Seq ベースの質問生成手法を提案し，ルールベースや統計的機械翻訳ベースの手法よりも生成精度が向上することを示している．Duan ら [6] は，質問生成を質問パターン，トピックといった要素に分解し，それらを推定するニューラルネットワークを実装することによる質問生成技術を提案している．Tang [22] らは，質問生成で生成した文を機械読解で回答するというプロセスを行うマルチタスク学習により，SQuAD や MSMARCO など複数の機械読解データセットにおいて，単純に質問生成をしたときよりも，より読解内容に適合した質問が生成できることを報告している．

質問応答や機械読解における質問生成では，入力としてパッセージと回答を与え，回答に対応する質問を生成することを目的としている．本論文では，回答は与えずパッセージと，質問文を与えたときに内容を具体化した質問を生成することを目的としているため，これらのタスクとは異なっている．

6.2 情報検索におけるクエリ拡張に関する研究

情報検索では，入力キーワードの変換や追加を行うクエリ拡張技術として研究が盛んに行われている．Nogueira ら [13] は，ニューラルネットワークと強化学習を組み合わせて，入力クエリに対して，関連文書数が最大となるようなクエリに変換する手法を提案している．Dehghani ら [4] は，クエリを一連のシーケンスとみなして，Seq2Seq モデルで次に推薦するクエリを生成する手法を提案している．このとき，クエリのキーワードにアテンションをかけ，推薦クエリのキーワードを Copy するか生成するかを決定している．情報検索においては，入力はキーワード組であることが一般的であるが，Song ら [19] は自然言語の質問を検索に最適なキーワード組に変換する手法を提案している．

本論文が対象とする質問応答は全て自然言語でやりとりが行われるため，クエリのキーワードに用いられる内容語だけでなく，機能表現等も正しく生成する必要がある．自然言語を扱うことができる検索システムでは，クエリを入力質問，検索結果のスニペット情報をパッセージと見なすことで，より具体的なクエリを作成するといった応用が可能であると考えられる．

6.3 タスク対話におけるスロットフィリングに関する研究

ユーザの情報要求や質問意図を明確化していくタスクでは，予約システムなどで用いられるタスク対話システムがある [26]．タスク対話では，場所や時間など，情報を提供するために必要な情報をスロットとして定義し，ユーザにスロットに関する情報を埋めるための質問を繰り返し行うことで，最終的な情報提供を行う．近年では，ニューラルネットワークを用いて，質問内容を決定する研究が多く報告されている [25, 27]．

本論文の提案手法は，予めスロットを用意することなく，パッセージからユーザの質問意図を明確化できそうな情報を自動抽出して，RQ を生成しているという違いがある．

7. おわりに

本論文では，質問応答の際に，ユーザの質問意図を明確化する目的で，質問内容を具体化する改訂質問生成（RQG）について述べた．機械読解と言語生成のニューラルネットワークを組み合わせ，パッセージ内で入力質問に関連している情報をニューラルネットワーク上で発見し，その情報を基に改訂質問（RQ）を生成することで，入力質問を具体化しつつ，知識源であるパッセージの内容に沿った質問を生成できる．

今後はより多様な改訂質問を生成するための手法や，英語などの多言語化を行い SQuAD などの機械読解のオープンデータにも本手法を適用することを検討している．

文　献

[1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[2] C. Buck, annis Bulian, M. Ciaramita, W. Gajewski, A. Gesmundo, N. Houlsby, and W. Wang. Ask the right questions: Active question reformulation with reinforcement learning. *Proc of the 6th International Conference on Learning Representations (ICLR2018)*, 2018.

[3] Z. Cao, C. Luo, W. Li, and S. Li. Joint copying and restricted generation for paraphrase. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI2017)*, pp. 3152–3158, 2017.

[4] M. Dehghani, S. Rothe, E. Alfonseca, and P. Fleury. Learning to attend, copy, and generate for session-based query suggestion. *Proc of the 2017 ACM on Conference on Information and Knowledge Management (CIKM2017)*, pp. 1747–1756, 2017.

[5] X. Du, J. Shao, and C. Cardie. Learning to ask: Neural question generation for reading comprehension. *Proc of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pp. 1342–1352, 2017.

[6] N. Duan, D. Tang, P. Chen, and M. Zhou. Question generation for question answering. *Proc of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP2017)*, pp. 866–874, 2017.

[7] J. Gu, Z. Lu, H. Li, and V. O. Li. Incorporating copying mechanism in sequence-to-sequence learning. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL2016)*, pp. 1631–1640, 2016.

[8] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *Proc of the 28th International Conference on Neural Information Processing Systems (NIPS2015)*, pp. 1693–1701, 2015.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[10] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. Opennmt: Open-source toolkit for neural machine translation. *Proc of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations (ACL2017)*, pp. 67–72, 2017.

[11] V. Kumar, S. Joshi. Incomplete follow-up question resolution using retrieval based sequence to sequence learning. *Proc of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2017)*, pp. 705–714, 2017.

[12] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *Proc of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, pp. 1412–1421, September 2015.

[13] R. Nogueira and K. Cho. Task-oriented query reformulation with reinforcement learning. *Proc of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP2017)*, pp. 574–583, 2017.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. *Proc of the 40th Annual Meeting on Association for Computational Linguistics (ACL2002)*, pp. 311–318, 2002.

[15] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *Proc of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pp. 1532–1543, 2014.

[16] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *Proc of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pp. 2383–2392, 2016.

[17] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, 1997.

[18] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *Proc of 5th International Conference on Learning Representations (ICLR2017)*, 2017.

[19] H.-J. Song, A.-Y. Kim, and S.-B. Park. Translation of natural language query into keyword query using a rnn encoder-decoder. *Proc of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2017)*, pp. 965–968, 2017.

[20] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *Proc of Deep Learning Workshop, ICML2015*, 2015.

[21] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Proc of the 27th International Conference on Neural Information Processing Systems(NIPS2014)*, pp. 3104–3112, 2014.

[22] D. Tang, N. Duan, T. Qin, and M. Zhou. Question answering and question generation as dual tasks. *arXiv*, abs/1706.02027, 2017.

[23] O. Vinyals and Q. V. Le. A neural conversational model. *Proc of the ICML Deep Learning Workshop 2015*, 2015.

[24] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou. Gated self-matching networks for reading comprehension and question answering. *Proc of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pp. 189–198, 2017.

[25] J. D. Williams, K. Asadi, and G. Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *Proc of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pp. 665–677, 2017.

[26] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li. Building task-oriented dialogue systems for online shopping. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI2017)*, pp. 4618–4625, 2017.

[27] T. Zhao, A. Lu, K. Lee, and M. Eskenazi. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. *Proc of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL2017)*, pp. 27–36, 2017.

[28] 西田京介, 斉藤いつみ, 大塚淳史, 浅野久子, 富田準二. 情報検索とのマルチタスク学習による大規模機械読解. 言語処理学会第24回年次大会論文集 (NLP2018), 2018.