抽出型要約との同時学習による回答根拠を提示可能な機械読解

西田 光甫[1] 西田 京介[1] 永田 昌明[2] 大塚 淳史[1] 斉藤 いつみ[1] 浅野 久子[1] 富田 準二[1]
[1] 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
[2] 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
nishida.kosuke@lab.ntt.co.jp

1 はじめに

テキストを読み解いて質問に答える機械読解が注目を集めており，SQuAD [7] を始めとして数多くのタスクが提案されている．SQuAD は 1 テキストに明記されている内容について質問する抽出型のタスクであり，実サービスでの利用に向けては様々な課題が残る．特に，質問の対象が 1 テキストに限定されている点は実サービスでは強い制約となるため，複数のテキストに書かれた記述を組み合わせて推論する multi-hop 型の機械読解タスク QAngaroo (WikiHop) [12], HotpotQA [13] が提案されている．

multi-hop 型の機械読解タスクは，テキスト中の複数の箇所に書かれた記述を根拠とする．そのため，タスクとしての難しさに加え，根拠部分を全て確認しないと回答の妥当性をユーザが理解できないことに課題が存在する．

そこで本研究では，図 1 に示すように，システムが質問へ回答すると共に回答の根拠となる文を抽出することを目指す．機械読解タスクで根拠文を抽出することによって，(1) 実サービスで利用する際に回答の妥当性やなぜ回答に至ったかといったユーザの解釈性が向上する (2) 文の重要性の情報を機械読解モデル内で暗に利用することで回答精度が向上する，の 2 点を期待することができる．

本研究では，機械読解における文章からの根拠文の抽出が，質問に応じて複数テキストの内容を要約するクエリ依存型要約タスク [4, 6] の一種であることに着眼した．本研究の貢献は以下の 2 点である．

- 機械読解と抽出型要約をマルチタスク学習することで，回答時に高精度に根拠文を抽出するモデル Query Focused Extractor (QFE) を提案する．QFE では，抽出する根拠文の数を入力に応じて適応的に決定できる．
- 評価実験により，QFE が HotpotQA [13] において state-of-the-art の性能を持つことを示した．

2 タスク設定

本研究では図 1 で示すシステムを構築する．そのため，次の定義を持つタスクに取り組む．

定義 1 (タスク)**.** 以下の入力を受けて出力を予測する．

入力 文章 P（テキストの集合），質問文 Q（テキスト）

出力 回答タイプ T（ラベル），回答 A（テキスト），根拠文 S（テキストの集合）

ユーザ： おじいちゃんの癌が見つかったのですが私の保険で入院費用をもらえますか？
システム： はい．
根拠 1： 二親等以内の親族の放射線治療は入院給付金日額の 10 倍を補償します．
根拠 2： 放射線治療は癌の主要な治療法のひとつである．

図 1 機械読解システムのイメージ

文章 P は長さや件数に制限を持たない．**質問文** Q は文章 P の内容に関する質問である．システムは質問文 Q に対して回答タイプ T や回答 A によって回答する．**回答タイプ** T は本研究では「はい・いいえ・抽出」の 3 つのラベルから構成される．**回答** A は回答タイプ T が「抽出」であるときのみ存在し，文章 P から抽出できるテキストである． **根拠文** S は文章 P 中の文のうち，質問文 Q に回答するために必要となる文の集合である．

3 提案手法

3.1 モデルアーキテクチャ

提案モデルのアーキテクチャの概要を下記に示す．根拠抽出層を除き，Yang らが HotpotQA [13] におけるベースラインモデル [3] として用いた構成と同じである．図 2 に概略図を示す．

入力： システムは文章 P と質問文 Q を文字列として受け取る．文章 P が複数のテキストから構成されていた場合，全てを連結した文字列とみなす．

単語理解層： 文章 P と質問文 Q を単語系列に変換し，ベクトル系列に変換する．単語ベクトルは単語由来のベクトルと文字由来のベクトルを繋げたベクトルである．単語由来のベクトルは事前学習済みの単語埋め込みを用いる．文字由来のベクトルは学習可能な文字埋め込みを CNN と max-pooling によって変換して得る [5]．出力は $P_1 \in \mathbb{R}^{l_w \times d_w}, Q_1 \in \mathbb{R}^{m_w \times d_w}$ である．ただし，l_w は文章 P の単語数，m_w は質問文 Q の単語数，d_w は単語ベクトルの大きさである．

文脈理解層： 入力 P_1, Q_1 を双方向 RNN によって変換した $P_2 \in \mathbb{R}^{l_w \times 2d_c}, Q_2 \in \mathbb{R}^{m_w \times 2d_c}$ を出力する．d_c は単方向 RNN の出力するベクトルの大きさである．

関係理解層： 入力 P_2, Q_2 を双方向アテンション [9]，双方向 RNN，セルフアテンション [11] を用いて変換したベクトル系列 $P_3 \in \mathbb{R}^{l_w \times d_c}$ を出力する．

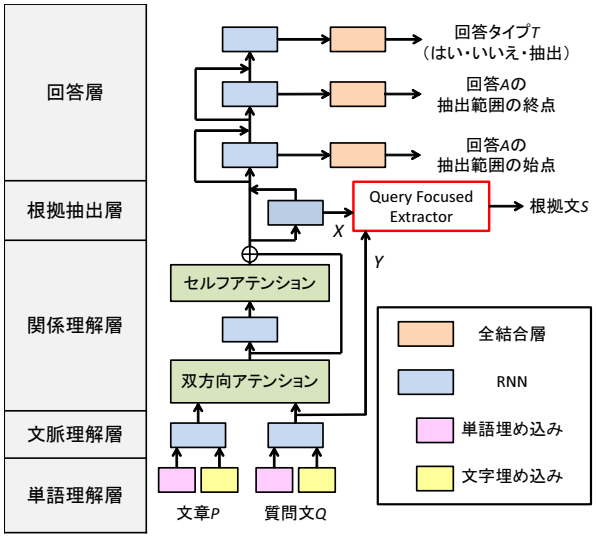

図 2　機械読解システムの全体像

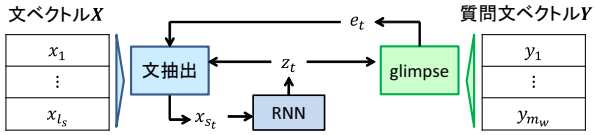

図 3　Query Focused Extractor の概要

根拠抽出層： 入力 P_3 を双方向 RNN によってベクトル系列 $[\overrightarrow{P_4}; \overleftarrow{P_4}] \in \mathbb{R}^{l_w \times 2d_c}$ に変換する．ここで，文 i の始点の単語を $j_1(i)$，終点の単語を $j_2(i)$ としたとき，文 i のベクトル x_i を

$$x_i = [\overrightarrow{p_{4,j_2(i)}}; \overleftarrow{p_{4,j_1(i)}}]$$

と定義する．後述する QFE では，文章 P の文数を l_s として，文レベルのベクトル系列 $X \in \mathbb{R}^{l_s \times 2d_c}$ と文脈理解層の出力 Q_2 を入力として受け取る．QFE は各文が根拠である確率分布

$$\Pr(i) = \mathrm{QFE}(X, Y = Q_2) \tag{1}$$

を出力する．最後に，単語レベル・文レベルのベクトルを結合し，$P_5 \in \mathbb{R}^{l_w \times 3d_c}$ を得る．

$$p_{5,j} = [p_{3,j}; x_{i(j)}]$$

なお，$i(j)$ は，単語 j が含まれる文の添字である．

回答層： 3 層の双方向 RNN によって P_5 を変換する．それぞれの RNN の出力は全結合層および softmax 操作により，回答 A の抽出範囲の始点の確率分布 $\hat{A}_1 \in \mathbb{R}^{l_w}$，終点の確率分布 $\hat{A}_2 \in \mathbb{R}^{l_w}$，回答タイプ T の確率分布 $\hat{T} \in \mathbb{R}^{|T|}$ に変換される．回答層は回答 A の抽出範囲と回答タイプ T のラベルを予測する．回答タイプ T が抽出以外の場合は回答 A が存在しないため，始点と終点の出力は無視される．

目的関数： 目的関数を $L = L_A + L_S$ としてマルチタスク学習する．ここで，L_A は回答に関する損失，L_S は根拠文抽出に関する損失である．回答の損失 L_A は，回答層の 3 つの確率分布 $\hat{A}_1, \hat{A}_2, \hat{T}$ のクロスエントロピー損失の和である．L_S の詳細は 3.3 節に示す．

3.2　Query Focused Extractor

本稿では図 2 中の Query Focused Extractor (QFE) として，クエリを用いない要約モデルである Chen と Bansal の抽出型文章要約モデル [1] を拡張したモデルを提案する．Chen と Bansal はアテンションを利用して，要約元文章の重要な部分を包含するように文を抽出した．QFE ではクエリ依存性を持たせるため，アテンションをクエリに，文抽出を文章 P に対して行う．つまり，クエリへのアテンションに基づいてクエリの各単語に関連する部分が含まれるように，文章 P 中の文を抽出する．QFE の概略は図 3 に示す．

QFE の入力は，文章 P の意味を表すベクトル系列 $X \in \mathbb{R}^{l_s \times 2d_c}$ と，質問文 Q の意味を表すベクトル系列 $Y \in \mathbb{R}^{m_w \times 2d_c}$ である．

根拠文を 1 つ抽出する操作を 1 時刻と定義し，抽出モデルの状態 z_t を RNN によって更新する．

$$z_t = \mathrm{RNN}(x_{s_t}) \in \mathbb{R}^{2d_c}$$

ただし，$s_t \in \{1, 2, \cdots, l_s\}$ は時刻 t に選ばれた文の添字である．時刻 t までに選ばれた文の集合を $S_t = \{s_1, s_2, \cdots, s_t\}$ と書く．時刻 t に第 i 文を確率分布

$$\Pr(i; S_{t-1}) = \mathrm{softmax}_i(u_i^t)$$

$$u_j^t = \begin{cases} v_p^\top \tanh(W_{p1}x_j + W_{p2}e_t + W_{p3}z_t) & (j \notin S_{t-1}) \\ -\infty & (\text{otherwise}) \end{cases}$$

に従って選び，$s_t = i$ とする．e_t は時刻 t における重要性を考慮した質問文ベクトルであり，glimpse ベクトル [10] と定義する．

$$a_j^t = v_g^\top \tanh(W_{g1}y_j + W_{g2}z_t)$$

$$\alpha^t = \mathrm{softmax}(a^t) \in \mathbb{R}^{m_w}$$

$$e_t = \sum_j \alpha_j^t W_{g1} y_j \in \mathbb{R}^{2d_c}$$

なお，RNN の初期値は X を affine 変換したベクトル系列を max pooling したベクトルとする．全ての W, v は訓練可能なパラメータである．

3.3　学習

訓練時は Teacher-Forcing によって文を抽出し，目的関数を最小化する．根拠文に関する損失は，負の対数尤度関数に coverage 機構 [8] による正則化を行う．

$$L_S = -\sum_{t=1}^{|S|} \log \left(\max_{i \in S \setminus S_{t-1}} \Pr(i; S_{t-1}) \right) + \sum_i \min(c_i^t, \alpha_i^t)$$

なお，第 1 項は抽出する根拠文の順序を任意とするために，最も選ばれやすい文から抽出する max 操作を行っている．coverage ベクトル c^t は $c^t = \sum_{\tau=1}^{t-1} \alpha^\tau$ である．終了時刻は訓練時は教師データの根拠文数である．テスト時は終了条件を満たした時刻とする．

抽出の終了条件を学習するために，ダミーの文（EOE 文と呼ぶ）を挿入し，EOE 文が QFE に選択された際に抽出を終了する．EOE 文に相当する文ベクトル $x_{EOE} \in \mathbb{R}^{2d_c}$ は学習可能パラメータとして全データで共通とし，全ての根拠文を出力した後に EOE

	文章 P の段落数	文章 P の単語数	質問文 Q の単語数	根拠文 S の文数
平均	10.0	1162.0	17.8	2.4
最大	10	3079	59	8
最小	2	60	7	2

表 1　HotpotQA 統計情報（開発データ，distractor 設定）

文を抽出するように Teacher-Forcing で訓練する．テスト時は EOE 文を出力した時点で抽出を終了する．

4　評価実験

4.1　データセット

本研究では，推論の根拠となる文の教師データを持つ multi-hop 型の機械読解タスクとして， HotpotQA [13] を利用する．HotpotQA は定義 1 を満たすデータセットである．

HotpotQA では，Wikipedia の 2 つの段落からの推論が必要な質問文 Q がクラウドソーシングにより作成されている．回答 A は，回答タイプ T が「抽出」の場合上記 2 段落から 1 つの範囲が選択される．回答の根拠となる根拠文 S の文数は任意（平均 2.4 文）である．文章 P は 10 段落であり，その内容に関しては 2 つの設定がある．**distractor 設定**では，質問文 Q の作成に用いた 2 段落に加えて，Wikipedia の全段落から質問文 Q をクエリとした bi-gram TF-IDF によって検索された 8 段落を利用する．**fullwiki 設定**では，10 段落全てを検索された結果とする．このため，fullwiki 設定は本来必要な 2 段落を含まない場合があり，その場合回答 A や根拠文 S が文章 P に含まれない．両設定は文章 P が異なるのみであり、評価における回答 A・根拠文 S は同じデータを用いる．よって，fullwiki 設定では，理想のモデルであっても 100%の精度とはならない．また，fullwiki 設定の場合も，訓練データは distractor 設定のデータが用いられる．distractor 設定の統計情報を表 1 に示す．

4.2　実験設定

比較手法として，3.1 節で述べた Yang らのベースラインモデル [13] を用いる．ベースラインモデルは図 2 中の QFE に対応する (1) 式で

$$\Pr(i) = \mathrm{sigmoid}(w^\top x_i + b)$$

としたモデルである．ただし，$w \in \mathbb{R}^{2d_c}, b \in \mathbb{R}$ は学習可能なパラメータである．

実験は NVIDIA Tesla P100 を 4 枚用いて行った．実装には Pytorch を用いた．全ての手法で $d_c = 150$, dropout の keep ratio を 0.8，バッチサイズを 72，学習率を 0.001 とした．QFE では RNN に GRU [2] を用いた．根拠抽出のデコード時の beam size を 2 とした．ベースラインモデルでは根拠文の抽出を決定する閾値として，根拠文抽出の部分一致が最も高い 0.4 を用いた．上記以外のハイパーパラメータは Yang らと同じ値を用いた．

実験では Yang らの評価指標 [13] に基づき，回答タイプ T・回答 A・根拠文 S の予測精度を評価した．回答，根拠文抽出ともに完全一致（EM）と部分一致（F1）を評価した．回答は回答タイプ T の一致で評価し，抽出の場合は回答 A の一致でも評価する．根拠文抽出の部分一致は抽出された文 id の真の根拠文 id への一致で測った．そのため，単語レベルでの部分一致は考慮されない．また，回答と根拠の精度双方を考慮した指標として Joint EM，Joint F1 [13] を用いる．

	回答		根拠文		Joint	
	EM	F1	EM	F1	EM	F1
Yang ら [13]	45.6	59.0	20.3	64.5	10.8	40.2
QFE	**53.9**	**68.1**	**57.8**	**84.5**	**34.6**	**59.6**

表 2　Test Set での結果（distractor 設定）

	回答		根拠文		Joint	
	EM	F1	EM	F1	EM	F1
Yang ら [13]	24.0	32.9	3.86	37.7	1.85	16.2
QFE	**28.7**	**38.1**	**14.2**	**44.4**	**8.69**	**23.1**

表 3　Test Set での結果（fullwiki 設定）

	回答		根拠文		Joint	
	EM	F1	EM	F1	EM	F1
Yang ら [13]	44.4	58.3	22.0	66.7	11.6	40.9
ベースライン	52.7	67.3	38.0	78.4	21.9	54.9
QFE	**53.7**	**68.7**	**58.8**	**84.7**	**35.4**	**60.6**
glimpse 未使用	53.1	67.9	58.4	84.3	34.8	59.6

表 4　Dev Set での結果（distractor 設定）

	回答		根拠文		Joint	
	EM	F1	EM	F1	EM	F1
Yang ら [13]	24.7	34.4	5.28	41.0	2.54	17.7
ベースライン	28.5	38.2	8.89	**45.5**	5.28	22.9
QFE	**29.0**	**38.8**	**14.4**	44.8	**8.43**	**23.3**
glimpse 未使用	28.6	38.2	13.9	44.5	8.30	23.0

表 5　Dev Set での結果（fullwiki 設定）

4.3　評価

テストデータにおける実験結果は，distractor 設定の結果を表 2 に，fullwiki 設定の結果を表 3 に示す．distractor 設定，fullwiki 設定ともに，QFE はベースラインモデルを大きく上回り，state-of-the-art の精度を達成した．特に根拠文の完全一致は distractor 設定で 37.5 ポイント（+185%），fullwiki 設定で 10.3 ポイント（+268%）と大きく向上している．

開発データでの distractor 設定における実験結果を表 4 に示す．なお，開発データでのベースラインは我々の追実装であり，ハイパーパラメータの違いにより精度が論文値から向上している．ハイパーパラメータを同一にした場合でも，QFE はベースラインモデルを全ての指標で上回った．また glimpse 操作を用いずに RNN による文抽出だけを行う手法も，全ての指標で QFE が上回ることを確認した．

開発データでの fullwiki 設定における実験結果を表 5 に示す．ハイパーパラメータを同一にした場合，QFE はベースラインモデルを根拠文抽出の F1 を除く指標で上回った．QFE が glimpse 操作を用いずに RNN による文抽出だけを行う手法を全ての指標で上回っていることを確認した．根拠文抽出の F1 でベースライン

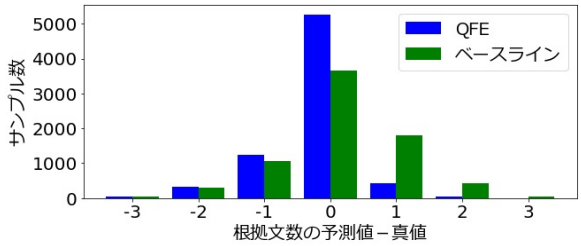

図 4　予測された根拠文の数から真の根拠文の数を引いた値.（distractor 設定）

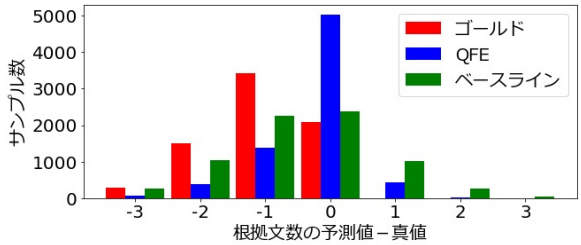

図 5　予測された根拠文の数（fullwiki 設定）から真の根拠文の数（distractor 設定）を引いた値.

から精度が悪化した理由は後に議論する.

4.4　議論

本小節では以下のリサーチクエスチョンを議論する.

根拠文の抽出は回答の精度向上に貢献するか. 表 4 に示す我々が追実装した 3 つの手法は，根拠抽出層以外は全て同じモデルを用い，また根拠の抽出結果を回答層で明示的に利用しないが，根拠の抽出精度が高い順に回答精度も高くなっている．これは，QFE のマルチタスク学習によって下層の RNN が回答にも資する特徴量を獲得するように訓練された結果であると解釈できる．また，表 4 の結果より，glimpse が精度向上に貢献していることが分かる.

QFE の高精度の要因は何か. ベースラインモデルは文章中の各文を独立に判定するため，図 4 が示すように，真の根拠文に対して過剰に根拠文を予測する場合がある．しかし，QFE は根拠文を逐次的に出力するため，過剰な出力が抑制できる．実際に，根拠文抽出の適合率は QFE が 88.4，ベースラインが 79.0 であり，再現率の QFE で 83.2，ベースラインで 82.4 に比べて大きな差となっている．以上のことから，QFE は根拠文の過剰な出力を抑制することができ，EM や適合率が高くなることを確認できた.

QFE の fullwiki 設定における課題は何か. fullwiki 設定は回答に必要な段落が文章 P に含まれない場合があるため，根拠文 S の抽出が不可能な場合がある．よって，図 5 では根拠文数の予測値 – 真値が零または負の値を出すことが理想（ゴールド条件）となる．それぞれの手法で図 4 と比較すると，ベースラインモデルでは実際に負の値を出すサンプルが増えている．しかし，QFE は図 4 と図 5 で近い分布をしており，根拠文が不足していることが結果に反映されていない.

原因としては，訓練データが回答に必要な 2 段落を全て含む distractor 設定のデータであるため，根拠文の不足や回答不能性をモデルが学習できないことが挙げられる．ベースラインは文ごとに根拠文スコアを出す手法であるため，スコアに根拠文の不足が反映された．しかし，QFE は抽出文数も自動で決定するため，訓練データである distractor 設定の抽出文数への依存が大きいと考えられる．解決策としては，根拠文が不足している教師データを用意することがある.

5　おわりに

本稿では multi-hop 型の機械読解タスクで根拠文を出力することに取り組み，要約タスクからの類推によって QFE を提案した，本研究の重要性を以下に示す.

- QFE はマルチタスク的に既存モデルに追加することが可能であり，広い汎用性を持つ.
- HotpotQA を用いて根拠の抽出と回答精度の関係について初めて詳細に考察し，回答根拠を提示可能な機械読解の実現に向けた貢献が大きいと考える.

今後の課題として，fullwiki 設定で根拠文の不足を認識することが挙げられる．実用面を考えると，将来的には根拠文の不足だけでなく，回答不能性を判定できるようになることが望ましい.

さらには，別の機械読解タスクで QFE を利用して，回答根拠を提示することが挙げられる．特に根拠文の教師データのないデータセットに対する弱教師あり学習・半教師あり学習を実現することで，QFE によってあらゆる機械読解タスクのデータセットで解釈性と精度を向上することを実現したい.

参考文献

[1] Y.-C. Chen and M. Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *ACL*, pp. 675–686, 2018.
[2] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pp. 1724–1734, 2014.
[3] C. Clark and M. Gardner. Simple and effective multi-paragraph reading comprehension. In *ACL*, pp. 845–855, 2018.
[4] H. T. Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, Vol. 2005, pp. 1–12, 2005.
[5] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pp. 1746–1751, 2014.
[6] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran. Diversity driven attention model for query-based abstractive summarization. In *ACL*, pp. 1063–1072, 2017.
[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *ACL*, pp. 2383–2392, 2016.
[8] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL*, pp. 1073–1083, 2017.
[9] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.
[10] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. In *ICLR*, 2016.
[11] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, pp. 189–198, 2017.
[12] J. Welbl, P. Stenetorp, and S. Riedel. Constructing datasets for multi-hop reading comprehension across documents. *TACL*, 6:287–302, 2018.
[13] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pp. 2369–2380, 2018.