E-001

トピック情報とアテンションつきの再帰型ニューラルネットを用いた 文章読解問題の解法の検討
TMA Net: Deep Neural Network for Reading Comprehension Combining Topic Information and RNN with Attention

髙島 侑里† 青野 雅樹†
Yuri Takashima Masaki Aono

1. はじめに

文章読解問題を計算機に行わせる人工知能技術が注目を浴びている．文章を与え，その内容に関連した質問に解答するシステムの正答率を測定することで，計算機の文章読解能力(Reading Comprehension)を測ることが可能である．

近年，文章とそれに対する質問，解答の 3 組を 1 問とした大規模なデータセットが公開された[1][2][3][4][5]．これらはどれもオンラインニュース記事や書籍などのテキストを利用することで構築されている．図 1 に CNN/Daily Mail QA データセット[1]の例を示す．ニュース記事(Raw Article)を文章(Document)，ニュース記事の要約(Raw Highlight)を利用して質問(Question)およびその解答(Answer)が作成されており，ニュース記事の要約文内の固有名詞の 1 つをプレースホルダとして伏せ，その部分に対応する固有名詞を文章から推測する穴埋め問題となっている．すべての固有名詞は匿名化されており，固有名詞に関する知識の有無による解答への影響を抑え，文章を理解する能力に焦点を当てている．

これらの大規模かつ質の高いデータセットにより，計算機による読解問題へのアプローチとして教師あり機械学習の実現が可能となり，特に大規模な訓練データを必要とする深層学習を利用した手法が発展を遂げている．Hermann らは LSTM[6]を利用した深層学習による手法が，従来のルールベースや Word Distance を用いた手法より有効であることを示した[1]．現在の読解問題に解答する効果的なアプローチは 2 種類に大別できるといえる．1 つ目は文章と質問を複数回交互に読む手法（マルチホップ手法）である[7][8][9]．2 つ目は機械翻訳に有効とされるアテンション技術[10]を活用し，質問内容に関連する文章の一部分に着目する手法（アテンション手法）である[1][11]．このマルチホップ手法とアテンション手法の両者を補完的に組み合わせた手法が，Dhingra らが提案する GA Reader である[12]．GA Reader ではアテンション技術により，質問と密接に関係している文章の着目すべき部分を明らかにし，文章と質問の解読を複数回行うことで高い正答率を達成している．さらに GA Reader では文単位[7][13]，単語単位[1][2][11][14]のアテンションとは異なる，意味レベルのアテンションを提案している．

本研究ではマルチホップ手法とアテンション手法に加え，文章が属するトピックに着目した手法の計 3 つを組み合わせた読解問題の解法モデルを提案する．同じデータセットの文章でも，文章が表す分野やトピックは多様である．例えば，CNN/Daily Mail QA データセットはニュース記事を利用しているが，その内容は政治，芸能，スポーツなど多岐にわたる．文章が属するトピックを詳細に分析し，その情報を利用することで文章の理解への貢献が期待できるといえる．

図 1 CNN/Daily Mail QA データセットの一例

本稿では CNN/Daily Mail QA データセットの中の CNN QA データセットを用いて，データセット内の文章が表すトピックについて調査を行う．さらにマルチホップ手法，アテンション手法と，トピックに着目した手法を組み合わせた読解問題の解法モデルを提案し，従来手法との比較実験を実施する．2 節では Reading Comprehension に関する論文を紹介する．3 節ではトピックモデルおよび CNN QA データセットのトピックについて詳述する．4 節ではマルチホップ手法，アテンション手法およびトピックに着目した手法を合体した読解問題の解法モデルの概要を説明する．5 節では提案モデルと従来手法による比較実験を実施し，その考察を述べる．6 節は結論および今後の課題について述べる．

† 豊橋技術科学大学, Toyohashi University of Technology

2. 関連研究

近年，Reading Comprehension に関する大規模なデータセットが複数公開されたことにより，Reading Comprehension の研究は盛んに行われている．ここでは計算機の文章理解力を測るためのデータセットの詳細および提案された解法モデルについて紹介する．

2.1 Reading Comprehension に関するデータセット

CNN/Daily Mail QA データセット：Hermann らは CNN [1] および Daily Mail [2] のニュース記事とその要約文を用いて自動で構築したデータセットを公開した[1]．文章および質問に存在する全ての固有名詞は共参照を保ったまま"@entity0", "@entity1"といったようなランダムな変数表現に置換されていることが特徴的な点である（図 1 の Document 参照）．この処理により，タスクを固有名詞に関する事前知識を必要としない純粋な読解問題に帰着させることができる．データサイズは，CNN について訓練用データが約 38 万問（約 9 万記事），開発用およびテスト用データが各々約 3 千問である．Daily Mail について訓練用データが約 88 万問（約 22 万記事），開発用データが約 6 万問，テスト用データが約 5 万問である．

Children's Book Test データセット：Hill らは電子図書館サイトの 1 つである Project Gutenberg [3] から入手した書籍のテキストを用いて構築したデータセットを公開した[2]．文章と質問文は，物語中の連続する 21 文から作成されている．そのうち，末尾の 1 文を除いた 20 文を文章に，末尾の 1 文を質問に割り当てる．そして質問文中の任意の 1 単語（固有名詞，名詞，動詞，前置詞）をプレースホルダとし，正解となる単語を文章および質問中に現れる同一の品詞の単語から選ばれた 10 個の解答候補から 1 つを選択する．解答候補がマスクされていない点が CNN/Daily Mail QA データセットと異なる．データサイズは，訓練用データが約 67 万問（98 冊），開発用データが 8 千問，テスト用データが 1 万問である．

Stanford Question Answering Dataset (SQuAD)：Rajpurkar らは Wikipedia の記事を用いて SQuAD1.0 を構築した[4]．Wikipedia の記事 536 件からクラウドソーシングを利用して約 10 万問の質問を生成している．さらに Rajpurkar らは文章に関連するものの，その文章を読んだだけでは答えられない質問を既存のデータセットに追加した SQuAD2.0 を公開した[5]．SQuAD1.0 の根本的な問題として挙げられていた「解答が必ず文章中に存在している」という点を解消し，より現実的で難易度が高いタスクとなっている．

2.2 Reading Comprehension のための解法モデル

Hermann らは文章dと質問qを結合したコンテキスト$g(d,q)$と解答候補の類似度を計算することで正解の推定を行っている[1]．$g(d,q)$の表現方法として，Deep LSTM Reader, Attentive Reader, Impatient Reader の 3 種類を提案している．Deep LSTM Reader では文章と質問を区切り記号で合体させ，順方向の LSTM に入力している．Attentive Reader は質問文を用いて文章中の各単語にアテンションを行うことで$g(d,q)$が作成される．Impatient Reader は質問文の単語ごとにアテンションを更新し，文章表現を逐次的に計算している．

Chen らは Hermann らの Attentive Reader を改良したモデルを提案している[11]．質問文を用いた文章へのアテンションの計算手法として双線形アテンション(bilinear attention)を採用している．双線形アテンションにより，内積によるアテンションよりも質問と文章の単語の類似度を柔軟に計算することができると示している．

Kadlec らは文章と質問を双方向の GRU ユニット[14]に入力することでそれぞれをベクトル表現に変換し，質問を用いた文章へのアテンションを行っている[15]．さらに，文章内で共通のエンティティごとにアテンションで出力された確率を足し合わせるポインターサム・アテンション(pointer-sum attention)を適用することで，文章内に 2 回以上出現するエンティティの確率を 1 つに集約する AS Reader を提案している．

Seo らは文字レベル，単語レベル，コンテキストレベルの分散表現を組み合わせ，質問を用いた文章へのアテンションと，文章を用いた質問へのアテンション(bi-directional attention)を使用した階層的なモデルを構築している[16]．CNN/Daily Mail QA データセットと SQuAD の 2 種類のデータセットで評価実験を行い，提案手法の有効性を確認している．

Yu らは畳み込みとセルフ・アテンション(self-attention)[17]を用いたモデルを構築し，SQuAD と Trivia QA[18]の 2 種類のデータセットによる実験の結果，従来手法より高い精度が得られることを示した[19]．Reading Comprehension タスクの解法モデルで広く用いられている再帰型ニューラルネットワーク(RNN)の代わりに，畳み込みとセルフ・アテンションを採用することで，正答率の向上だけでなく，訓練モデルを構築する速度の大幅な短縮を実現している．

3. トピック抽出手法および CNN データセットのトピック分析

本節では，提案モデルの鍵となるトピック抽出手法についてその詳細を説明する．また，CNN データセットを適用して構築したトピックモデルについて具体例を交えながら分析を行う．

3.1 トピック抽出手法

QA で用いられる文章および質問にはそれぞれ何らかのトピックが存在すると考えられる．例えば CNN/Daily Mail QA データセットはニュース記事を用いて構築されており，ニュース記事は政治，芸能，スポーツなど，多様なトピックが存在する．そのためこれらのトピック情報を事前知識として取り入れることで，解答の推定に貢献できると考えられる．本研究ではトピックモデルの 1 つである潜在的ディリクレ配分法(LDA: Latent Dirichlet Allocation)[20]を用いてトピックの抽出を行う．LDA は 1 つの文書が潜在的に複数のトピックを含んでいると仮定し，文書を複数のトピックの混合分布で，各トピックを複数の単語の混合分布で表現したものである．

次に，LDA による CNN QA データセットのトピック抽出手法について説明する．トピックモデルの構築には，文章，質問，解答の 3 組からなる訓練用データ約 38 万問を

1 https://edition.cnn.com
2 http://www.dailymail.co.uk/ushome/index.html
3 https://www.gutenberg.org/

表 1　属する文章数が多いトピック上位 15 件と，そのトピックが構成する単語

トピック番号	トピック単語	属する文章数
5	year, team, said, world, players, time, football, game, club, season	34444
17	said, people, government, killed, group, police, attack, attacks, security, for$es	29916
3	second, first, win, match, th, two, goal, last, home, final	26830
6	show, film, music, like, nt, movie, one, said, also, new	25649
19	said, court, case, charges, attorney, trial, judge, prison, years, according	21933
21	said, president, government, country, minister, told, would, statement, also, official	21250
2	nt, like, one, would, know, people, think, get, time, going	19971
23	said, people, plane, flight, according, storm, water, passengers, one, two	19478
1	said, family, told, father, home, mother, year, son, old, death	19310
22	company, new, million, said, companies, business, also, users, use, market	15960
0	people, says, nt, children, school, work, help, life, women, many	15594
8	military, war, forces, weapons, would, conflict, international, regime, troops, region	15354
10	said, report, information, officials, investigation, government, security, statement, tol	15121
4	president, government, political, economic, country, economy, years, would, power, many	15113
7	said, would, president, campaign, bill, sen, nt, health, state, federal	13205

使用し，その中の文章のみを用いる．トピックが普遍的な単語で表現されるのを避けるため，前処理として文章内の単独の記号，数字および“a”, “the”などのストップワードは除去した．ストップワードの選定は NLTK[4] の stopwords[5] を参考にしたさらに，訓練用データ全体で出現回数が 10 回未満の単語と匿名化された固有名詞群(“@entity”)はノイズとなるため削除した．生成されるトピック数は 25 とし，文書あたりのトピック数を制御するパラメータαは 1/25 に設定した．

3.2 CNN データセットのトピック分析

3.1 節で構築したトピックモデルを用いて，トピックの分析を行った結果を示す．CNN QA データセットの文章についてそれぞれトピック解析を行い，トピック確率が最も高いトピックを，その文章が属するトピックと仮定した．表 1 は属する文章数が多いトピック（上位 15 件）と，そのトピックが構成する単語（上位 10 単語）を表している．表 1 から政治，スポーツ，軍事，天気などのトピックが存在することが確認でき，CNN ニュースの記事は多様なトピックで構成されていることが分かる．特にスポーツに関するニュース（トピック 5，トピック 3），国の情勢に関するニュース（トピック 17）が多数存在している．

トピック 21 とトピック 4 の 2 つに着目すると，両者は政治に関するトピックであると考えられる．さらに分析すると，トピック 21 は大統領や大臣の発言について，トピック 4 は経済に関するトピックであることが推測できる．このトピックモデルは「政治」などの広いトピックに限らず，その中でトピックを細かくカテゴライズできることが確認でき，より詳細なトピック分析が可能であることが分かる．

4. トピック抽出と深層学習を組み合わせた読解問題の解法モデル

本節では文章と質問を複数回交互に読む手法（マルチホップ手法）と，質問内容に関連する文章の一部分に着目する手法（アテンション手法），文章および質問のトピックに着目した手法の 3 つを組み合わせた読解問題の解法モデルを提案する．具体的には，Dhingra らが提案するマルチホップ手法とアテンション手法の両者を補完的に組み合わせた GA Reader[10]をベースとし，文章と質問の持つトピック情報を単語の分散表現と結合する．本研究で提案する読解問題の解法モデルの概要図を図 2 に示す．以下，解法モデルについて詳述する．

Embedding 層：初めに，文章dおよび質問qを単語に分割したあと，各単語をe次元の分散表現に変換する．分散表現へのマッピングにより，式(1)と式(2)に示すような文章と質問についてD_E, Q_Eを得る．ただし，mは文章の単語数，nは質問の単語数とする．

$$D_E = [x_1, x_2, \dots, x_m] \quad x_1, x_2, \dots, x_m \in \mathrm{R}^e \tag{1}$$

$$Q_E = [y_1, y_2, \dots, y_n] \quad y_1, y_2, \dots, y_n \in \mathrm{R}^e \tag{2}$$

LDA 層：Embedding 層と同様に文章および質問を単語に分割した後，3.1 節で述べた LDA モデルにより各単語のトピック確率を算出する．そのため，文章，質問について式(3)と式 (4) に示すようなトピック情報を含んだベクトル表現を得る．ただしtはトピック数である．

$$D_T = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m] \quad \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m \in \mathrm{R}^t \tag{3}$$

$$Q_T = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n] \quad \tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n \in \mathrm{R}^t \tag{4}$$

RNN 層：双方向 GRU[13]を用いて文脈を考慮した分散表現への変換を行う．文章の各単語について，Embedding 層，LDA 層で得られた文章のベクトル表現を連結した$D = [d_1, \dots, d_m] = [x_1 \parallel \tilde{x}_1, \dots, x_m \parallel \tilde{x}_m]$を順方向，逆方向の GRU に入力することで式(5)と式(6)に示すような局所文脈の分散表現$\overrightarrow{h_i}, \overleftarrow{h_i}$を得る．ただし，$\parallel$は連結演算を表す．

$$\overrightarrow{h_i} = \mathrm{GRU}(\overrightarrow{h_{i-1}}, d_i), i = 1, \dots, m \tag{5}$$

$$\overleftarrow{h_i} = \mathrm{GRU}(\overleftarrow{h_{i+1}}, d_i), i = m, \dots, 1 \tag{6}$$

そして，順方向と逆方向の GRU で得られた分散表現を連結した$h_i = \overrightarrow{h_i} \parallel \overleftarrow{h_i}$を，$i$番目の単語における文脈を考慮し

4 http://www.nltk.org/
5 https://pythonspot.com/nltk-stop-words/

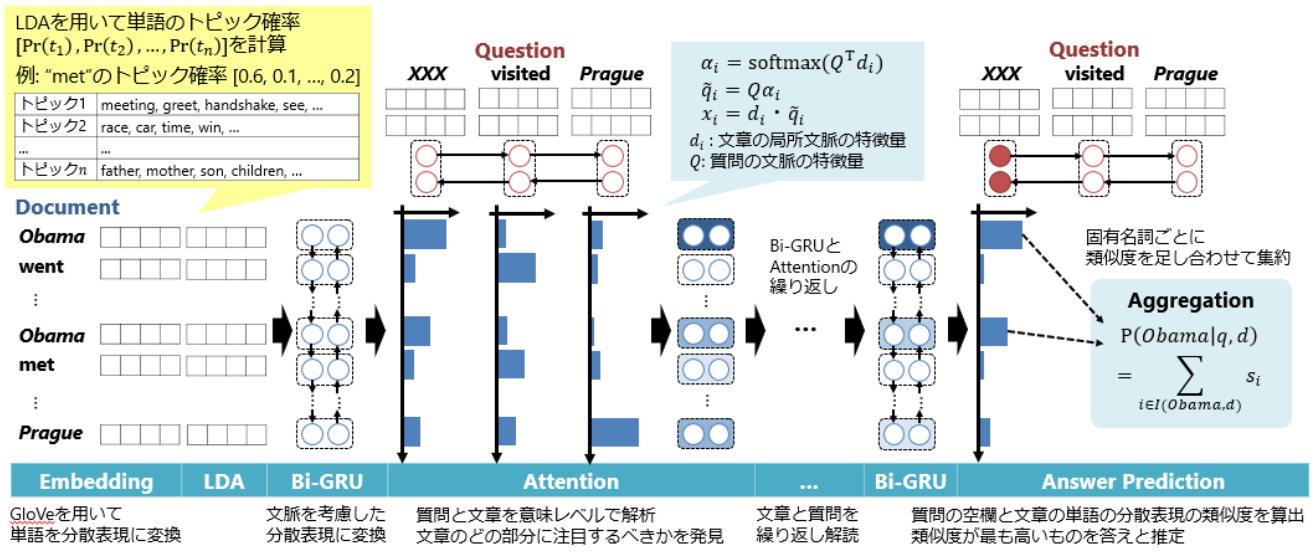

図 2　提案手法の概要図

た分散表現と定義する．質問に関しても同様の手法で局所文脈の分散表現Qを得る．

Attention 層：アテンションの方法は Dhingra らが提案した Gated-Attention Module[12]を参考にしている．式 (7)で文章における局所文脈表現d_iと質問のアテンションを行い，式 (8)で文章の局所文脈を考慮した質問の分散表現を得る．そのため，文章の各局所文脈に応じた，m種類の質問の分散表現が出力されることとなる．さらに，式 (9)でアテンションにより得られた質問の分散表現を文章の分散表現に掛けることで，質問の内容を考慮した文章の分散表現を新たに生成する．⊙は，要素ごとの積演算を表す．この分散表現をもう一度 RNN 層，さらに Attention 層に入力することで，文章と質問を繰り返し解読する．本研究では Dhingra らの予備実験の結果に従い， RNN 層と Attention 層を 2 回繰り返し，最後の Answer Prediction 層を含め合計 3 回 GRU ユニットへの入力を行った．

$$\alpha_i = \mathrm{softmax}(Q^T d_i) \tag{7}$$

$$\tilde{q}_i = Q\alpha_i \tag{8}$$

$$\tilde{d}_i = d_i \odot \tilde{q}_i \tag{9}$$

Answer Prediction 層：質問文中のプレースホルダの位置をlとし，文頭からプレースホルダまでの局所文脈$\overrightarrow{h_l}$と文末からプレースホルダまでの局所文脈$\overleftarrow{h_l}$を，GRU を用いて分散表現にエンコードする．その後，2 つの分散表現を連結して$h_l = \overrightarrow{h_l} \| \overleftarrow{h_l}$を得る．次に，式(10)でプレースホルダに関する局所文脈の分散表現h_lと，RNN 層と Attention 層を通して得られた文章の分散表現$\tilde{D}$の内積を計算し，softmax 関数を施して文章の局所文脈とプレースホルダの局所文脈の類似度を表す確率分布を得る．

$$s = \mathrm{softmax}(h_l^T \tilde{D}) \tag{10}$$

次に，文章dと質問qの解答候補$c \in C$について，cが答えとなる確率を式 (11)で定式化する．ただし，$I(c,d)$は文章d内での解答候補cの出現位置である．それぞれの解答候補が文章内で複数存在する場合（図 2 では，たとえば Obama）もあるため，それらの確率を足し合わせ 1 つに集約する処理を行っている．この処理は Kadlec らが提案したポインターサム・アテンション(pointer sum attention) [15]を参考にしている．最後に，最も確率が高い解答候補cを式 (12)により答えと推定する．

$$\Pr(c|d,q) \propto \sum_{i \in I(c,d)} s_i \tag{11}$$

$$a^* = \mathrm{argmax}_{c \in C} \Pr(c|d,q) \tag{12}$$

5. 比較実験

提案手法の有効性を確認するために，CNN QA データセット[1]を用いて評価実験を行った．5.1 節では，詳細な実験設定を述べる．5.2 節では従来手法との比較実験の結果を述べ， 成功例を交えながら考察を行う．

5.1 実験設定

Embedding でマッピングする単語は，エンティティ，プレースホルダと出現回数が高い単語の合計 5 万語に制限し，その他の単語は未知語とした．Embedding の次元数は 100 とし，初期値として訓練済みの GloVe[21]を利用した．GloVe に存在しない単語については，[0.0,1.0]の一様分布で初期値を与えた．トピック数は予備実験の結果より 25 に設定した．

モデルの学習について，最適化手法は Adaptive Moment Estimation (Adam) [22]を採用し，学習率は 0.001 とした．誤

6 https://github.com/keras-team/keras

差関数をクロスエントロピーとし，クロスエントロピーが最小となるように学習を行った．バッチサイズは 32 に設定した．ドロップアウト率は 0.2 とし，双方向 GRU ユニットの入力ベクトルに適用した．GRU の中間層の次元は 128 とした．なお，本モデルは Keras[6] によって実装した．

5.2 実験結果と考察

CNN QA データセットを用いた実験結果を表 2 に示す．CNN QA データセットのテスト用データにおける本モデルの正答率は 78.7%であった．他の手法と比較すると，最も精度の高い Dhingra らが提案した GA Reader[12]より正答率が向上しており，本手法の有効性が確認できる．また他の単体モデルだけでなく，アンサンブルモデルと比較しても本手法の精度が他のアンサンブルモデルの精度を上回っていることがわかる．

表 2　CNN QA データセットの各モデルによる正答率

Model	Dev	Test
Deep LSTM Reader [1]	55.0	57.0
Attentive Reader [1]	61.6	63.0
Impatient Reader [1]	61.8	63.8
AS Reader (single) [15]	68.6	69.5
Stanford AR (single) [11]	73.8	73.6
BiDAF [16]	76.3	76.9
GA Reader [12]	**77.9**	77.9
Stanford AR (ensemble) [11]	77.2	77.6
AS Reader (avg ensemble) [15]	73.9	75.4
提案手法	**77.9**	**78.7**

次に CNN QA データセットのテスト用データの中で，提案手法で解答の推定に成功した具体例を示す．図 3 は GA Reader で答えの推定に失敗したのに対し，提案手法で正解した問題である．なお，データセット内で固有名詞は“@entity”で匿名化されているが，図 3 では実際のニュース記事[7]をもとに固有名詞を復元している．図 3 で示した例はドラマの放送が再開されることに関して報じたものである．質問のプレースホルダに対し，GA Reader は“Fox”（テレビ放送ネットワーク）と推定し，提案手法では“Brian Grazer”（映画プロデューサー）と正しい解答を得た．CNN QA データセットでは固有名詞が匿名化されているため，文章中の”Arrested Development”（テレビドラマ）や“Netflix”（オンライン DVD レンタル及び映像ストリーミング配信事業会社）といった固有名詞の事前知識を活用することはできない．そのため，その他の単語の情報が重要となり，“producer”, “streamed”, “episodes”といった特徴的な単語のトピック情報からドラマに関するトピックであることを推定し，その情報を利用することで正解を導き出すことができたと考えられる．

6. おわりに

本論文では，計算機が文章読解能力を測る手法として QA タスクに着目し，マルチホップ手法，アテンション手法に加えて，文のトピック情報に着目した手法を組み合わせた複合的なモデルを構築した．トピック情報の抽出には，代表的なトピックモデルである LDA を利用し，CNN QA データセットを用いて実際にモデルを構築して分析を行った．分析の結果，CNN QA データセットは政治，天気，スポーツ，軍事など様々なトピックが存在することが確認できた．また構築した LDA モデルと深層学習を適用し，QA タスクを解く提案モデルを作成した．CNN QA データセットで実験を行った結果，他の手法の精度を上回り，提案手法の有効性を確認した．

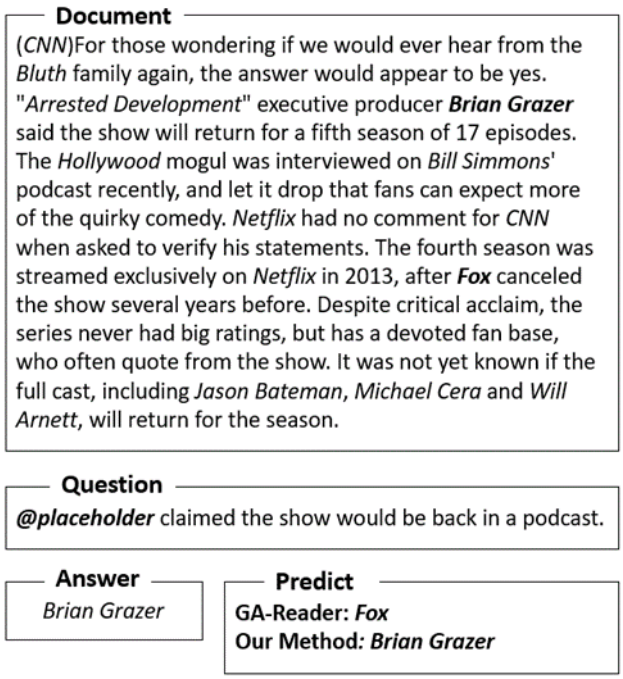

図 3　提案手法での成功例

今後の課題として，一つ目にトピックモデルの改良が挙げられる．LDA モデルを構築する際に使用する単語にストップワードや出現回数による制限をかけたものの，“said”がほとんどのトピックに出現していたり，“nt”などの否定語の一部が現れたりした．そのため TF-IDF などで単語の重みづけを行うことで，あらゆる文書で横断的に出現する単語を取り除き，より特徴的な単語に焦点を当てたトピックモデルの構築を目指したい．

二つ目に他のデータセットによる提案モデルの有効性の確認が挙げられる．本研究では CNN QA データセットを用いたが，同じくニュース記事を利用して構築された Daily Mail QA データセットでの実験を行いたい．さらに，Reading Comprehension を測るデータセットである Children's Book Test データセットや SQuAD での有効性の確認も望ましい．

最後に，新規手法による更なる提案モデルの改良が挙げられる．Chen らは古典的手法によるモデルの実験で，N-gram 特徴量の有効性を示している[11]．そのため畳み込みニューラルネットワークなどを利用して，N-gram のような前後の文脈を強調した特徴量をニューラルネットで実現させ，新規モデルの構築を目指したい．

7 https://edition.cnn.com/2015/04/07/feat-arrested-development-return/index.html

謝辞

本研究の一部は，科研費基盤（B）（課題番号 17H01746）の支援を受けて遂行した．

参考文献

[1] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, “Teaching Machines to Read and Comprehend”, Advances in Neural Information Processing Systems, pp. 1684–1692 (2015)
[2] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston, “The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representations”, International Conference on Learning Representations (2016)
[3] Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester, “Who did What: A Large-Scale Person-Centered Cloze Dataset”, Empirical Methods in Natural Language Processing, pp.2230-2235 (2016)
[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text”, Empirical Methods in Natural Language Processing, pp.2383-2392 (2016)
[5] Pranav Rajpurkar, Robin Jia and Percy Liang, “Know What You Don't Know: Unanswerable Questions for SQuAD”, Association for Computational Linguistics (2018)
[6] Sepp Hochreiter and Jurgen Schmidhuber, “Long Short-Term Memory”, Neural Computation, pp.1735–1780 (1997)
[7] Jason Weston, Sumit Chopra, and Antoine Bordes, “Memory Networks”, International Conference on Learning Representations (2015)
[8] Alessandro Sordoni, Phillip Bachman, Adam Trischler, and Yoshua Bengio, “Iterative Alternating Neural Attention for Machine Reading”, arXiv preprint arXiv:1606.02245 (2016)
[9] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen, “ReasoNet: Learning to Stop Reading in Machine Comprehension”, KDD, pp.1047-1055 (2017)
[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate”, International Conference on Learning Representations (2015)
[11] Danqi Chen, Jason Bolton, and Christopher D Manning, “A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task”, Association for Computational Linguistics, pp.2358-2367 (2016)
[12] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov, “Gated-Attention Readers for Text Comprehension”, Association for Computational Linguistics, pp.1832-1846 (2017)
[13] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus, “End-To-End Memory Networks”, Advances in Neural Information Processing Systems (2015)
[14] 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”, Empirical Methods in Natural Language Processing (2014)
[15] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst, “Text Understanding with the Attention Sum Reader Network”, Association for Computational Linguistics, pp.908-918 (2016)
[16] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi, “Bidirectional Attention Flow for Machine Comprehension”, International Conference on Learning Representations (2017)
[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, “Attention Is All You Need”, Neural Information Processing Systems (2017)
[18] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer, “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension”, Association for Computational Linguistics, pp.1601-1611 (2017)
[19] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi and Quoc V. Le, “QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension”, International Conference on Learning Representations (2018)
[20] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, “Latent Dirichlet Allocation”, Journal of Machine Learning Research, pp. 993-1022 (2003)
[21] Jeffrey Pennington, Richard Socher, and Christopher Manning, “GloVe: Global Vectors for Word Representation”, Empirical Methods in Natural Language Processing, pp.1532–1543 (2014)
[22] Diederik P. Kingma and Jimmy Lei Ba, “Adam: a Method for Stochastic Optimization”, International Conference on Learning Representations (2015)