Vol.2019-NL-241 No.3
2019/8/29

対話システムが積極的な情報提供をするための推薦知識獲得

福原 裕一[1,a)]　水野 淳太[1]　門脇 一真[1,2]　飯田 龍[1,3]　鳥澤 健太郎[1,3]

概要：対話システムで「ステーキをおいしく焼くにはお肉は常温に戻しておくことをお勧めします」や「ウール素材のお手入れにはブラシをまめに行うことを勧めます」といったユーザに役立つ情報を提供するためには，このような推薦対象（例「ステーキをおいしく焼く」）と推薦情報（例「お肉は常温に戻す」）から成る推薦知識を大規模に獲得することが重要となる．本研究では，まず Web 文書から推薦知識の候補として抽出した最大 2 文を対象に，推薦対象とその対象に関する推薦情報が含まれるか否かを BERT を用いて分類する手法を開発した．さらに，この手法で得られた推薦知識を含む文を対話システムを通じてコンパクトにユーザに提示するために，推薦知識を要約する手法を pointer-generator network を用いて開発した．これらの手法を学習・評価するために，推薦知識分類のためのデータとして 58,978 件，推薦知識要約のためのデータとして 19,647 件を人手でアノテーションして作成した．評価実験の結果，推薦知識分類の性能として精度約 72%，推薦知識要約の性能として ROUGE-2 F 値で約 76%を得た．

1. はじめに

近年，ユーザからの多様な自然言語による要求，依頼に応えるインタフェースとして対話システムが注目されており，Google Home，Siri，Amazon Echo といった対話システムが普及しつつある．これらの多くの商業的対話システムでは堅実に応答が可能な天気に関する情報の提供や，スマートデバイスと連携させることでそれを操作するといったユーザ要求への応答が主であり，雑談的な応答をオープンなドメインで発話させることは非常に難しいと考えられる．一方で，いわゆる雑談を行うことを目的とした対話システムの開発も進んでおり，例えば我々が研究開発している対話システム WEKDA [21]（図 1 に動作画面を示す）は，入力されたユーザ発話から質問応答システム WISDOM X [15][*1]で許容可能な質問（ユーザが関心を持ちそうな情報を提供できそうな質問）を自動生成し，適切な応答を返すことができる．今後は，雑談などを通じてユーザに役立つ情報をより積極的に提供していくことが求められると考えている．例えば，ユーザが「ステーキが美味しく焼けなくて…」などと自身が困っている状況をシステムに伝えた時に，システムが「ステーキをおいしく焼くにはお肉は常温に戻しておくことを勧めます」といった，その状況に対して有用な情報を返すといった対話を実現したい．

図 1　対話システム WEKDA の動作例

このようなユーザが発話したトラブル等に関する発話に

[1] 情報通信研究機構データ駆動知能システム研究センター
〒 619-0289 京都府相楽郡光台 3-5
[2] 株式会社日本総合研究所
〒 141-0022　東京都品川区東五反田 2 丁目 18 番 1 号
[3] 奈良先端科学技術大学院大学　先端科学技術研究科
〒 630-0192　奈良県生駒市高山町 8916 番地の 5
[a)] fukuhara@nict.go.jp
[*1] https://wisdom-nict.jp にて試験公開中

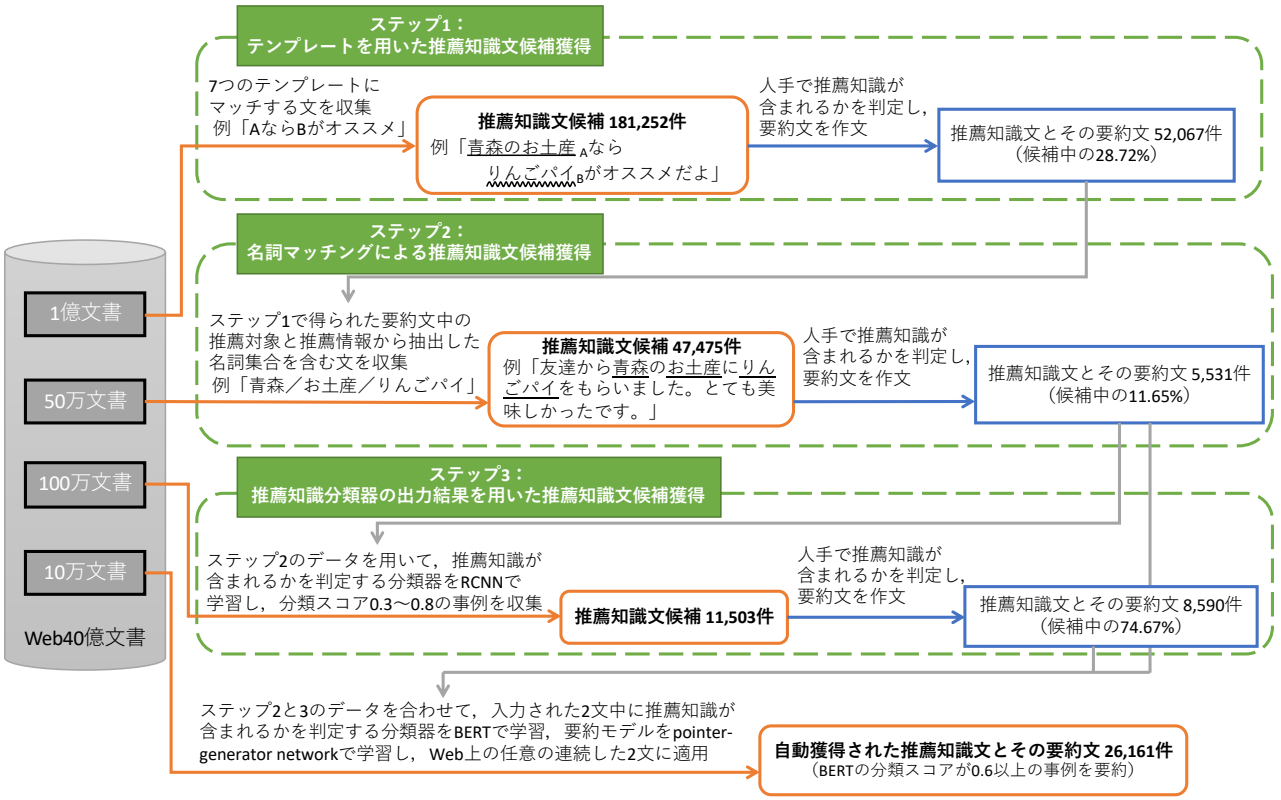

図 2 推薦知識獲得手法の全体図

関して適切にそのユーザに役立つ情報を提供するためには，大規模な Web 文書集合といった知識源からあらかじめユーザが抱えるトラブルなどの推薦の対象と，その対象に対してユーザに提示する推薦の情報を対で獲得し，それらを適切にユーザに対して応答することが考えられる．例えば，上述の「ステーキが美味しく焼けなくて」というユーザ発話に対してはシステムが応答するために「ステーキを焼くにはお肉を常温に戻しておくことをお勧めします」という推薦情報を事前に獲得しておくことで，適切な応答が可能である．ここでユーザが問題解決等を求める対象 (上の例では「ステーキが美味しく焼く」) を**推薦対象**，それに対して対話システムが応答すべきユーザに役立つ情報 (上の例では「お肉を常温に戻しておくこと」) を**推薦情報**と呼び，この 2 つの対を**推薦知識**と呼ぶこととする．なお，推薦対象，推薦情報のいずれも名詞句，述語句，文のいずれであっても良いこととする．

本研究では，我々がクロールした Web40 億文書から推薦情報をテンプレートに基づく抽出と単語単位のマッチングによる抽出の 2 種類の抽出方法を段階的に適用することで推薦知識獲得のための候補を獲得し，それらに人手でアノテーションを行い，学習データを効率よく作成することで推薦知識分類器を作成する．特に，推薦知識が書かれたテキストは Web 等においても比較的少なく，効率よく正例を見つけるために，上述したような 2 段階の学習データ作成を実施した．

さらに本研究では，Web 文書から抽出された推薦知識が記載された文（以降，**推薦知識文**と呼ぶ）は，推薦知識とは無関係な内容も含む場合が多く，対話システム等での活用時に問題がおきることが想定されるため，それらを除外し，コンパクトでなおかつ冗長でない推薦知識を推薦知識文から要約によって抽出する．ここで要約したものを**推薦知識要約**と呼ぶ．これを行うために，人手で推薦知識文に分類したものを対象に推薦知識要約を人手で作成し，要約用の学習データの作成も行った．

本研究を通じて作成した学習・評価用事例は推薦知識分類用の事例が 58,978 件，要約用の事例が 19,647 件であり，これらを使用して抽出，要約の評価実験を行った．分類には近年着目されている BERT[3] を利用し，分類性能として平均精度で約 94%を達成した．また，要約に関しては pointer-generator network[18] を利用して学習・評価を行い，ROUGE-2 F 値で約 71%という性能を得た．なお，後に述べるように，上記の評価実験で使われたデータには偏りがあるためと，Web40 億文書全体から得られる推薦知識の総数を見積もるために，さらなる評価実験を行なった．具体的には，Web40 億文書の一部である 10 万文書（これはこれまでの実験で使われていないもの）から任意の連続した 2 文を推薦知識文候補として抽出し，それらを対象に推薦知識分類器を適用して 26,161 件の推薦知識文を獲得した．このうち，500 事例をサンプリングして人手で評価を行ったところ，推薦知識の分類性能は精度で 72%，要約

A に (は)B を勧める	A に (は)B を推薦する
A に (は)B が良い	A に (は)B がお勧め
A に (は)B はいかが	A に (は)B は最適
A なら B がオススメ	

図 3 ステップ 1 で利用した 7 つのテンプレート

性能は ROUGE-2 F 値で約 76%であった．このことから，Web10 万文書からは約 18,000 件の推薦知識が獲得できることがわかる．さらに推薦知識分類器を 40 億文書全体に適用した場合には概算で約 7.2 億件（重複を含む）もの推薦知識を獲得可能である．

2. 段階的な推薦知識文候補の獲得

1 節で述べた推薦知識を効率的に収集するために，本研究では下記の 3 段階のデータ作成の方法を採用することで，効率的に推薦知識文の候補を獲得する．

ステップ 1 「A に B を勧める」等のいくつかのテンプレートをデータ中の各文と照合することで，推薦知識を含む可能性のある文を推薦知識文候補として獲得し，人手でそれら候補が適切な推薦知識が含まれるか否かの分類作業と推薦知識文の要約作業を行う．

ステップ 2 ステップ 1 で得られた推薦知識要約に含まれている名詞集合を獲得し，それを含む連続 2 文を推薦知識文候補として，人手で推薦知識を含むか否かの分類作業と推薦知識文の要約作業を行う．

ステップ 3 ステップ 2 の推薦知識の分類作業結果を用いて推薦知識文候補が推薦知識を含むか否かを判別する分類器を学習させ，その分類器を未知の Web 文書に適用，その分類結果を用いて作業対象を選別した上で分類と要約の作業を行う．

ステップ 1〜3 の概要を図 2 に示す．各ステップの詳細について以降の節で説明する．

2.1 ステップ 1 : テンプレートを用いた推薦知識文候補の獲得

我々が獲得したい推薦知識は典型的には「キウイを早く熟させるにはりんごやバナナとキウイをいっしょにビニール袋に入れるのがお勧め」のように「〜に (は)〜がお勧め」のようなテンプレートをともなって記述されると考えられ，かつ，このようなテンプレートをともなって記述されている場合は推薦知識である可能性が高いと考えられる．このため，まずはいくつかのテンプレートを事前に用意し，それを Web 文書中の文に適用することで推薦知識文候補を獲得する．

具体的には，図 3 に示す 7 つのテンプレートを用い，そのテンプレートと Web 文書中の各 1 文との照合を行い，合致した 1 文（対象文と呼ぶ）に加え，その前後 1 文，文書のタイトルを抽出し，推薦知識文候補とした．テンプレートと 1 文を照合する際は，テンプレートおよび照合対象となる文を MeCab[9]（JUMAN 品詞体系 [10]）で形態素解析し，テンプレート中の A および B に続く助詞と，テンプレート末尾の述語や名詞が，原形および品詞の両方でマッチしている場合にテンプレートにマッチしたと判断する．例えば，図 4(a) に示す例では，助詞の「に」「は」「が」および述語の「良い」が対象文に含まれているので，テンプレート「A に (は)B が良い」にマッチしたと判断する．

ステップ 1 ではテンプレートを用いた照合を Web40 億文書中の 1 億文書を対象に実施し，この結果，異なり数で 181,252 件の推薦知識文候補を獲得した．ここで得られた全事例を対象に推薦知識文候補が推薦知識を含むか否かを人手で分類した．この作業には 7 名のアノテータが従事し，各事例 1 名のアノテータが分類作業を行った．この結果，52,067 件の推薦知識を得た．さらに，推薦知識に分類された事例については追加でその分類を行ったアノテータが推薦知識文 (タイトル，対象文，前後 1 文) から推薦知識の要約を人手で記述した．人手で推薦知識を含むと分類された推薦知識文とその要約の具体例を図 4 の (a) と (b) に示す．

2.2 ステップ 2 : 名詞マッチングを用いた推薦知識文候補の獲得

ステップ 1 の推薦知識文候補の抽出ならびにアノテーションの結果，Web1 億文書から約 5 万件の推薦知識を獲得したが，この知識の件数は 1 億文書から獲得した量としては大きいとは言えない．規模が小さくなってしまっている主な原因は，Web 文書に適用したテンプレートが限定的であったことが考えられる．このため，テンプレートに依存しない名詞のマッチングを利用した推薦知識の自動獲得手法を提案する．

この手法では下記の手順で推薦知識文候補を獲得する．

(1) ステップ 1 で作成した各推薦要約から推薦対象部，推薦情報部に出現している名詞をそれぞれ抽出する．以降，推薦対象部から得られた名詞集合を**名詞 (推薦対象)**，推薦情報部から得られた名詞集合を**名詞 (推薦情報)** と呼ぶ．

(2) Web 文書内の隣接 2 文中にある推薦知識要約から得られた名詞 (推薦対象)，名詞 (情報) の一部が出現している場合に，その 2 文を推薦知識文候補として抽出する．

より厳密には，名詞 (推薦対象) と名詞 (推薦情報) に出現する名詞を 3 つ以上含む隣接 2 文を抽出する．ただし，名詞 (推薦対象) と名詞 (推薦情報) に含まれる名詞をそれぞれ 1 つ以上含むこととする．

推薦知識要約内の名詞に着目した理由としては，名詞に比べて動詞，形容詞などの述語は多様な書かれ方，言い換えをされる可能性があるため，それらを含めると照合がうまくいかなくなるためである．

推薦知識要約から名詞を抽出する際は，「こと」等の形

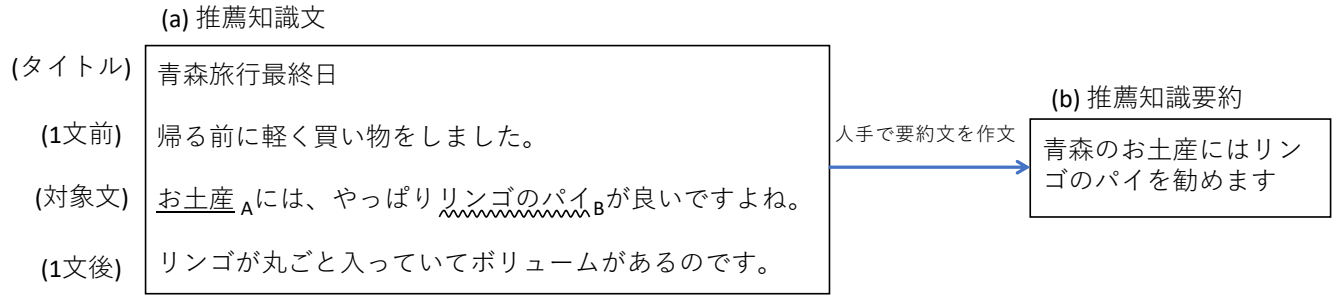

図 4 ステップ 1 で利用したテンプレートと収集された 4 文の例とその推薦知識要約（下線部と波線部がそれぞれテンプレートによって検出された推薦対象と推薦情報を表す）

式名詞は除外した．推薦知識要約内から名詞を特定する際は，JUMAN 品詞体系 [10] を利用した MeCab[9] を利用して形態素解析を行い，その結果を利用して名詞部分を特定した．

表 1 に実際にステップ 2 で抽出した推薦知識文候補を示す．この例では，推薦要約「炭水化物の餃子の皮で糖質を燃やしてスタミナにするには豚肉のビタミンＢ１、にんにくやにらのアリシンの餃子の具を勧めます」に出現する名詞「炭水化物」,「豚肉」等を含む隣接 2 文を Web 文書から抽出している．この例では図 3 のテンプレートとは合致しないが，内容としては「夏バテを解消するにはスタミナの素になるビタミンＢ１の宝庫の豚肉を勧める」といった内容を表しているため，推薦知識文として獲得することが望ましい．

ただし，推薦知識要約に含まれる名詞を含む隣接 2 文を網羅的に抽出した場合，Web 文書固有のノイズが含まれることになる．そこで日本語らしくないテキストを除くために平仮名 3 文字以上が含まれているテキストのみを収集した．これは平仮名が 3 文字未満のテキストからは文とはみなせないような 2 文（漢字，記号等だけで構成されているもの）しか抽出できなかったという予備調査の結果を反映して適用している．

ステップ 2 のアノテーションでは，上述の名詞 (推薦対象)，名詞 (推薦情報) の対をステップ 1 でアノテーションした 52,067 件の推薦知識要約から抽出し，それらを Web40 億文書中の 50 万文書に適用し，推薦知識文候補 47,475 件を抽出した．これら全ての事例に対して推薦知識か否かのアノテーションとを実施した．さらに，推薦知識に分類された場合は推薦知識要約の記述作業も行った．この作業には 13 名のアノテータが従事し，各事例に対して 3 名のアノテータが独立に推薦知識の分類と要約を行った．各事例の最終的な推薦知識か否かのラベルは 3 名のラベルの多数決で決定した．この結果，5,531 件の推薦知識を得た．推薦知識分類作業の一致率を Fleiss' Kappa[4] で調査したところ，Kappa 値は 0.533（中程度の一致）であった．

ただし，ステップ 2 の推薦知識要約では，ステップ 1 の場合と異なり事前にテンプレートとの照合によって推薦対象，推薦情報を特定した上でアノテータに提示することができない．このため，推薦知識要約を記述するアノテータは与えられた 2 文内の内容語を抜粋し，さらに助詞等の機能語を挿入することで推薦要約の作成を行った．2 文を用いて複数の推薦知識要約が作成可能な場合は，1) 推薦対象と推薦情報に共通する述語があるもの，2) 推薦対象・推薦情報の字数が少ないもの，3) 対話システムの発話として相応しい内容であるものという優先度にしたがい，作成する推薦知識要約を選択する．

2.3 ステップ 3 : 推薦知識分類器の出力結果を利用した推薦知識文候補の獲得

ステップ 1 のアノテーションではテンプレートを用いることで，アノテーションした総数のうち約 29%（52,067/181,252）が推薦知識であったのに対し，ステップ 2 のアノテーションでは約 11.3%（4,083/36,072）しか推薦知識となっておらず，正例の割合が少ない．

そこで，正例を増やすためステップ 2 でアノテーションした推薦知識文候補の一部の 36,072 件を用い，推薦知識の分類器を構築し，それを利用して推薦知識文候補を分類し，分類結果を参照してアノテーションする価値のある事例を選別した上で，推薦知識か否かの分類，また推薦知識要約のアノテーションを実施する．

具体的には，36,072 件（うち正例としての推薦知識は 4,083 件）の人手で要約したデータが集まった段階で，回帰型ニューラルネットワーク（RNN）と畳み込みニューラルネットワーク（CNN）を組み合わせた RCNN モデル [11] で学習させ，分類器を構築した[*2]．分類器の学習時の設定として，RNN には biLSTM を用い，RNN のユニット数は片方向当たり 100，CNN のフィルタサイズは 150，バッチサイズは 100，Adam 学習率は 1e-3 を用いた．アノテーション済みデータの約 1 割は検証データとして学習には用いず，early stopping のために使用した．また，この分類器の学習・評価では日本語 Wikipedia 全文を word2vec[14] の Skip-gram で学習させた 300 次元の word embeddings を

[*2] ステップ 3 実験時点では BERT[3] が公開されていなかった．

表 1　ステップ 2 で抽出した推薦知識文の候補

推薦知識要約	名詞 (推薦対象)	名詞 (推薦情報)	推薦知識文候補 (新規)
炭水化物の餃子の皮で糖質を燃やしてスタミナにするに は豚肉のビタミンＢ１、にんにくやにらのアリシンの餃子の具を勧めます	炭水化物，餃子，糖質，スタミナ	豚肉，ビタミンＢ１，にんにく，にら，アリシン，餃子	豚肉はビタミンＢ１の宝庫です。 ビタミンＢ１は夏バテの解消などスタミナの素になる大切な栄養素です。

「推薦知識文候補 (新規)」中の太字部分が名詞 (推薦対象) もしくは名詞 (推薦情報) に含まれる名詞を表す．

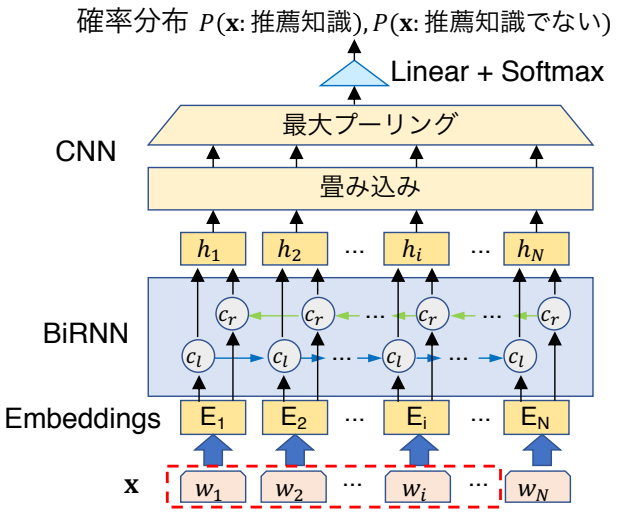

図 5　RCNN モデルの概要（$w_1, \cdots, w_N$ は推薦知識文候補に含まれる各単語を表す）

読み込んで利用した．RCNN モデルの概要を図 5 に示す．

次に，これまで利用してきた Web 文書とは重複しない Web100 万文書からステップ 2 と同様の手順にしたがい，かつ，ステップ 2 で利用した名詞 (推薦対象) と名詞 (推薦情報) を用いて，推薦知識文候補 107,062 件を抽出した．これらの事例に対して上述の RCNN モデルを適用し，分類スコアが 0.3〜0.8 の区間に含まれる事例のみを抽出した．この結果得られた 11,503 件の事例をステップ 3 のアノテーションに利用した．ここで，分類スコアが 0.3〜0.8 の区間に含まれる事例のみに着目したのは分類が難しい事例に関してアノテーションをすることによる能動学習的な分類性能の向上を意図したためである．これにより獲得できる推薦知識の量と種類は増えたが，同時に Web ページに比較的多く存在する不動産や求人に関係する情報を含む推薦知識を多く獲得することになった．例えば「宮城県でマンションをお探しの方には xxx 不動産を勧めます」や「芸能関係の求人には xxx 求人誌を勧めます」といったものである．これらは確かに推薦知識ではあるものの対話での使いどころの難しい事例である．

ステップ 3 のアノテーションではステップ 2 と同様に 13 名のアノテータが作業に従事し，各事例に対して 3 名のアノテータが独立に推薦知識分類，要約の作業を行った．ステップ 2 と同様に 3 名の多数決で推薦知識分類のラベルを決定した．この結果，8,590 件（ステップ 3 全体の 7 割）の推薦知識を得た．ステップ 3 の推薦知識分類の作業の一致率は Fleiss' Kappa 値で 0.584（中程度の一致）であった．

表 2　ステップ 1〜3 で得られた推薦知識文候補と推薦知識要約の件数

	推薦知識文候補	推薦知識　(%)
ステップ 1	181,252 件	52,067 件 (28.72%)
ステップ 2	47,475 件	5,531 件 (11.65%)
ステップ 3	11,503 件	8,590 件 (74.67%)

ステップ 1，2，3 の作業を通じてアノテーションされた事例数を表 2 にまとめる．

3. 評価実験

ステップ 2 と 3 で作成したデータ 58,978 件（うち正例としての推薦知識は 14,121 件で全体の 23.94%）を使い推薦知識分類を行う二値分類器を構築した．その際，推薦知識文の間で類似する事例が多かったため，推薦知識文の間の bag-of-words でのコサイン類似度を計算し，類似度が 0.7 以上となる事例を同一クラスとするように実験データを 10 分割した．具体的には，全データからランダムに事例を 1 つ選び，次に選んだ事例と最初の事例とのコサイン類似度が 0.7 以上ならその事例と同じクラスに，そうでなければクラスが 10 個になるまで新しいクラスを作りその中に入れた．クラス数が 10 になった後は，選んだ事例のコサイン類似度が 0.7 以上のクラスがあればそのクラスに割り当て，それ以外の場合は，ランダムに 10 クラスのいずれかに割り当てた．10 分割したデータの 3 クラスをそれぞれ検証，開発，評価に割り当て，残りの 7 クラスを学習データとした．表 3，4 に実験データの内訳を示す．

推薦知識分類の構築には，図 6 に示す BERT アーキテクチャ [3] を利用した．BERT は，Transformer[20] をベースとし，大規模なテキストで pre-training を行うことによって，数多くのタスクで state-of-the-art を達成したモデルである．

本実験では，BERT モデルの pre-training に，コーパスとして日本語 Web40 億文書から抽出した テキストを用いた．この 40 億文書から，CRF ベースの因果関係抽出器 [16] で因果関係（原因と結果の組）として分類された 1 文または 2 文と，その前後文脈で構成されたテキストパッセージ（最大 7 文）を 2.5 億件抽出し，pre-training には，そのうちランダムに選択した 280 万パッセージに含まれる 1,956 万文を用いた[*3]．Pre-training のハイパーパラメータとし

[*3] Pre-training には，Wikipedia や因果関係抽出器を用いる前の生の Web コーパスを利用することも考えられるが，予備実験において，これらよりも因果関係抽出器で抽出されたテキストのみ

表 3　推薦知識分類のデータの内訳

	学習	検証	開発	評価	合計
正例	10,457 (25.41%)	1,152 (20.68%)	1,180 (21.38%)	1,180 (23.17%)	14,121 (23.94%)
負例	31,684	4,419	4,338	4,416	44,857
合計	41,141	5,571	5,518	5,748	58,978

表 4　推薦知識要約のデータの内訳

学習	検証	開発	評価	合計
14,462	1,741	1,700	1,744	19,647

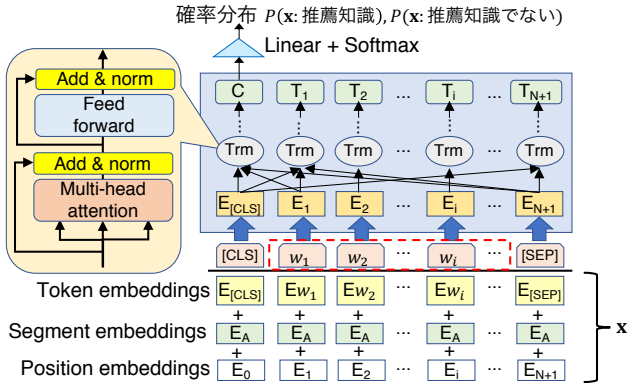

図 6　BERT の概要（$w_1, \cdots, w_N$ は推薦知識文候補に含まれる各単語を表す）

て，テキストの最大長は 128，バッチサイズは 50，語彙数は 10 万を用い，その他の設定は Devlin ら [3] の提案する $\text{BERT}_{\text{base}}$ と同様[*4]とした．また，学習は Google が公開している Tensorflow による実装[*5]をベースに用いたが，入力テキストは MeCab(JUMAN 品詞体系)[9], [10] で形態素に分かち書きしたものをそのまま用いるように変更した．また BERT への入力には，Devlin らと同様，単語を表す token embeddings，文の種類を表す segment embeddings，単語の位置を表す position embeddings の和を用いた．

BERT モデルの fine-tuning では，分類対象の推薦知識文候補に含まれる 2 文を segment embeddings 上では区別せず，同一のものとして BERT への入力とした．ハイパーパラメータとして，バッチサイズ 32，学習率 1e-5，2e-5，3e-5，4e-5，およびエポック数 1，2，3 の全組み合わせで実験を行い，開発データセットにおける平均精度が最高となるモデルを選択した．また，実験には，PyTorch の実装[*6]を本タスクにあわせて修正したものを用いた．

推薦知識要約においては，1 人以上が推薦知識要約を作成できた事例をもとに実験を行った．ここで，同一事例に関する最大 3 人の要約文の中から 1 件をランダムに選択したものを学習または正解データとして実験に用いた．

要約文の生成は，attention メカニズムを利用した seq2seq[2] の改良版である pointer-generator network[18] を用い，元文から要約に必要な表現をコピーし，その他の部分についても適宜生成して，自然な文を生成する．また，要約文の作成では，ビーム幅 3 で，最大単語長を 50 語に制限して探索を行った．また，デコード時に UNK が出力される場合，Jean ら [7] の手法に従い，入力文に出現している語に置き換えるように設定した．

実験は，エンコーダは 1，2，3 層の biGRU，デコーダは 1，2，3 層の GRU の全 9 通りの組み合わせで行い，開発データセットにおける ROUGE-1 F 値が最高となるモデルを選択した．その他のパラメータとして，Adam 学習率 1e-3，バッチサイズ 32 を設定し，また，日本語 Wikipedia 全文を word2vec[14] の Skip-gram で学習させた 500 次元の word embeddings をエンコーダとデコーダに読み込んで利用した．Pointer-generator network の実装は OpenNMT-py[8] の PyTorch 実装をベースにしたが，デコーダの初期値は，Iida ら [6] が提案するようにエンコーダ RNN の隠れ層の平均値で初期化をするように変更した．

ステップ 2，3 の実験の結果を表 5 (a) に示す．推薦知識分類の性能は，評価データセットにおいて平均精度 94.24% ，推薦知識要約の性能は，評価データセットにおいて ROUGE-2 （F 値）71.80% を得た．

4. Web 文書からの大規模推薦知識の獲得

前節で説明した実験結果は，特にステップ 2 で得られた学習データがステップ 1 で得られた推薦知識に含まれる名詞を含む文に限定されているため，精度が高めに出ている可能性がある．したがって，ステップ 2，3 で得られた学習データ全てを使って作った推薦知識分類器，要約器を未知の Web 文書に適用して，さらに精度を計測することを試みた．より具体的には，Web10 万文書から抽出した任意の連続した 2 文を入力とし，推薦知識と応答に利用できる文を獲得する．最終的には我々がクロールした大規模な Web データ（Web40 億文書）へ適用することを考えているが，手始めに Web10 万文書からどの程度推薦知識を獲得できるのかを確かめた．Web10 万文書から抽出した任意の連続した 2 文 130,749 件を 3 節で構築した分類器で判定し，そ

を用いる方が性能がよいことが確認できた．また，この文数は日本語 Wikipedia 全文に含まれる文数と同じとなるように設定した．

*4　12 層 Transformer，768 隠れ状態，3072 フィルタ，100 万学習ステップ数（1% warmup），Adam 学習率 1e-4.

*5　https://github.com/google-research/bert

*6　https://github.com/huggingface/pytorch-pretrained-BERT

表 5　推薦知識分類，要約の実験結果

データセット		推薦知識分類				推薦知識要約		
		再現率	精度	F 値	平均精度	ROUGE-1	ROUGE-2	ROUGE-L
(a) 表 3，4 のデータ	開発	86.02	87.95	86.98	94.58	79.64	71.01	77.98
	評価	86.86	87.01	86.93	94.24	80.14	71.80	78.48
(b) Web 10 万文書		–	72.00	–	–	83.86	76.84	82.92

(c) の Web10 万文書を対象にした評価では分類器が 0.6 以上のスコアを出力した事例の一部を評価し，
それ以外の事例についての正解が未評価なため，再現率，F 値，平均精度は計算できない．

の結果得られた推薦知識文を要約器で要約した．要約は分類器の出力する分類スコアが 0.6 以上の 26,161 件に対して行なった．

BERT を用いて構築した分類器の分類スコアが 0.6 以上の事例 26,161 件から 500 件をランダムサンプリングし，それを対象に第一著者が推薦知識分類と要約の作業を行った．この結果，360 件が推薦知識を含むと分類した．このことから，未知の Web 文書中の隣接 2 文に関する分類精度が 72%であることがわかる．つまり，Web10 万文書から得た 26,161 件の 72%（約 18,000 件）を推薦知識として獲得可能だと考えられる．表 6 は実際に獲得できた推薦知識の一部である．

さらに，この 360 件を入力とし pointer-generator network で要約した結果を人手要約の結果を使用し評価した結果，要約性能は表 5 (b) に示すように，ROUGE-2 F 値で 76.84%であった．360 件の推薦知識文 (2 文) の平均単語数は平均 55.63 単語であったが，推薦知識要約に要約することで平均 17.68 単語までコンパクトにすることができた．

3 節で示した分類の性能 (表 5(a)) は分類スコア 0.5 以上で精度を算出しているのに対し，本節で示した結果 (表 5(b)) では分類スコア 0.6 以上で精度を求めているため，直接比較することができない．そこで，実験結果 (a) に対しても分類スコア 0.6 以上として精度を求めたところ，90.88%であった．つまり，未知の Web 文書に適用した場合の精度 (72%) と比較して，約 19 ポイントもの差が生じている．このような差が生じた理由として，表 5(a) の実験に用いたデータはステップ 1 で得られた推薦知識要約中の名詞を含むという制約のもとで収集されており，学習・評価データに偏りがあるためだと考えられる．このため，今後このような学習データの偏りを軽減するデータ収集方法を検討する予定である．

Web10 万文書から抽出した任意の連続した 2 文 130,749 件（重複を除く）から獲得できる推薦知識は，約 18,000 件であるためこれを我々がクロールした大規模な Web データ（Web40 億文書）へ適用すると約 7.2 億件（重複を含む）の推薦知識を獲得できる試算になる．

5. 関連研究

ユーザの入力やプロファイルなどに基づいて，何らかの情報を推薦する研究は古くから行われているが，多くの研究でユーザに提供される情報は，映画や TV 番組 [5]，ニュース [1], [13]，SNS のハッシュタグ [12] といった，いわゆる商品やサービスが主である．この場合重要となるのは，レビューサイトなどからの評価文の抽出 [17] や，その評価の対象の同定である [22]．

推薦知識の一部には，問題とその対策が含まれるが，Varga ら [19] は，災害時に SNS に投稿される膨大なテキストからそれらを自動的に抽出する仕組みを開発した．例えば，「水が出なくなった」という問題や，「給水車が来た」といった対策を SNS 上（彼らは Twitter を利用）から自動的に抽出し，その対応付けも行っている．

一方で我々が実現したいのは，商品やサービスに限らず，ユーザが抱えている問題やおかれている状況（推薦対象）に対して，適切なアドバイス（推薦情報）を提供することであり，ドメインは限定されていない．さらに，ユーザに提供する推薦情報は，表 6 で示したように，要約モデルで推薦知識文を要約することで「○○サポート (を勧めます)」といったサービス名単体である場合や，「巻き始めが斜めになるようにすること (を勧めます)」といった文レベルの表現など，伝達すべき内容に応じて適切に調整することを目指している．このようなオープンドメインの内容をコンパクトな要約の形式で保持しておくことで，対話システム等で推薦すべき知識をユーザに提示する際に，幅広い情報を端的かつ適切に伝えることが可能となる．

6. おわりに

本研究では，Web 文書から推薦知識の候補として抽出した最大 2 文を対象に，推薦対象とその対象に関する推薦情報が含まれるか否かを BERT を用いて分類する手法を開発した．さらに，この手法で得られた推薦知識を対話システムを通じてコンパクトにユーザに提示するために，推薦知識を要約する手法を pointer-generator network を用いて開発した．これを Web10 万文書に適用し，約 18,000 件の推薦知識を得た．これを我々がクロールした大規模な Web データ（Web40 億文書）へ適用すると約 7.2 億件（重複を含む）の推薦知識を獲得できる試算になる．今後は，さらに推薦知識分類，要約技術を改善することで，さらに質が高く多様な推薦知識要約を獲得する予定である．さらに，この獲得した推薦知識要約を現在我々が開発を進めている音声対話システム WEKDA[21] に組み込むことで，よ

表 6　Web10 万文書から自動獲得した推薦知識要約の一部

BERT が推薦知識を含むと分類した 2 文	pointer-generator network による推薦知識要約の結果	人手評価
アスパラガスに豚肉を巻く時は、巻き始めが斜めになるようにするときれいに仕上がる。夏の揚げ物にはやっぱり辛口のビールが一番！	アスパラガスに豚肉を巻く時には巻き始めが斜めになるようにすることを勧めます	適切
豚肉に含まれているビタミンＢ１の量は、牛肉の１０倍もあるそうです。スタミナをつけたいときは、豚肉を食べるといいですね。	スタミナをつけたいときには豚肉を食べることを勧めます	適切
「今日も明日も、笑顔が輝くように。」看護師・保育士・介護職の転職サポートは○○○○サポート	看護師・保育士・介護職の転職サポートには○○○○サポートを勧めます	適切
ダンス衣装、ダンスウェア、アクセサリー、ドレス、衣装販売、通販、輸入。通販でアクセサリーをお探しなら海外のダンス衣装、ドレスを販売する「 ○○○○ ． ｃｏｍ 」へ	アクセサリーを探すには海外のダンス衣装、ドレスを販売することを勧めます	不適切

りユーザに役立つ情報を雑談などを通じてコンパクトに提供する予定である．

参考文献

[1] An, M., Wu, F., Wu, C., Zhang, K., Liu, Z. and Xie, X.: Neural News Recommendation with Long- and Short-term User Representations, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 336–345 (2019).

[2] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proceedings of the 3rd International Conference on Learning Representations* (2015).

[3] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186 (2019).

[4] Fleiss, J. L.: Measuring nominal scale agreement among many raters, *Psychological Bulletin*, Vol. 76, No. 5, pp. 378–382 (1971).

[5] Gomez-Uribe, C. A. and Hunt, N.: The Netflix Recommender System: Algorithms, Business Value, and Innovation, *ACM Transactions on Management Information Systems*, Vol. 6, No. 4, pp. 13:1–13:19 (2015).

[6] Iida, R., Kruengkrai, C., Ishida, R., Torisawa, K., Oh, J.-H. and Kloetzer, J.: Exploiting Background Knowledge in Compact Answer Generation for Why-Questions, *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 142–151 (2019).

[7] Jean, S., Cho, K., Memisevic, R. and Bengio, Y.: On Using Very Large Target Vocabulary for Neural Machine Translation, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1–10 (2015).

[8] Klein, G., Kim, Y., Deng, Y., Senellart, J. and Rush, A.: OpenNMT: Open-Source Toolkit for Neural Machine Translation, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pp. 67–72 (2017).

[9] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 230–237 (2004).

[10] Kurohashi, S., Nakamura, T., Matsumoto, Y. and Nagao, M.: Improvements of Japanese morphological analyzer JUMAN, *Proceedings of The International Workshop on Sharable Natural Language*, pp. 22–28 (1994).

[11] Lai, S., Xu, L., Liu, K. and Zhao, J.: Recurrent Convolutional Neural Networks for Text Classification, *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2267–2273 (2015).

[12] Li, Y., Liu, T., Jiang, J. and Zhang, L.: Hashtag Recommendation with Topical Attention-Based LSTM, *Proceedings of the 26th International Conference on Computational Linguistics*, pp. 3019–3029 (2016).

[13] Liu, J., Dolan, P. and Pedersen, E. R.: Personalized news recommendation based on click behavior, *Proceedings of the 15th international conference on Intelligent user interfaces*, pp. 31–40 (2010).

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119 (2013).

[15] Mizuno, J., Tanaka, M., Ohtake, K., Oh, J.-H., Kloetzer, J., Hashimoto, C. and Torisawa, K.: WISDOM X, DISAANA and D-SUMM: Large-scale NLP Systems for Analyzing Textual Big Data, *Proceedings of the 26th International Conference on Computational Linguistics (Demo Track)*, pp. 263–267 (2016).

[16] Oh, J.-H., Torisawa, K., Hashimoto, C., Sano, M., De Saeger, S. and Ohtake, K.: Why-Question Answering using Intra- and Inter-Sentential Causal Relations, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1733–1743 (2013).

[17] Reschke, K., Vogel, A. and Jurafsky, D.: Generating Recommendation Dialogs by Extracting Information from User Reviews, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 499–504 (2013).

[18] See, A., Liu, P. J. and Manning, C. D.: Get To The Point: Summarization with Pointer-Generator Networks, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1073–1083 (2017).

[19] Varga, I., Sano, M., Torisawa, K., Hashimoto, C., Ohtake, K. o., Kawai, T., Oh, J.-H. and De Saeger, S.: Aid is Out There: Looking for Help from Tweets during a Large Scale Disaster, *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1619–1629 (2013).
[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems 30*, pp. 5998–6008 (2017).
[21] 水野淳太，クロエツェージュリアン，田仲正弘，飯田　龍，呉　鍾勲，石田　諒，淺尾仁彦，福原裕一，藤原一毅，大西可奈子，阿部憲幸，大竹清敬，鳥澤健太郎：WEKDA：Web40 億ページを知識源とする質問応答システムを用いた博学対話システム，人工知能学会第 84 回言語・音声理解と対話処理研究会資料，pp. 135—-142 (2018).
[22] 林部祐太：宿レビューからの肯定的事実と推薦対象の抽出，言語処理学会第 25 回年次大会発表論文集，pp. 554–557 (2019).