DEIM Forum 2022 C41-1

質問生成と機械読解に基づく情報検索アルゴリズム

薄羽　皐太† 　加藤　誠†† 　藤田　澄男†††

† 筑波大学 情報学群 知識・情報図書館学類　〒 305–8550 茨城県つくば市春日 1-2
†† 筑波大学 図書館情報メディア系 / JST さきがけ　〒 305–8550 茨城県つくば市春日 1-2
††† ヤフー株式会社　〒 102–8282 東京都千代田区紀尾井町 1-3
E-mail: †s1811465@s.tsukuba.ac.jp, ††mpkato@acm.org, †††sufujita@yahoo-corp.jp

あらまし　本論文では，機械読解モデルを用いることによってアドホック検索タスクを解く方法を提案する．アドホック検索タスクを機械読解モデルで解くために，検索クエリからその背後にある質問を生成し，その質問に対する答えを含むかどうかを機械読解モデルによって判定することで，文書の適合性を推定する．機械読解モデルとして，機械読解タスクのデータセットである SQuAD2.0 でファインチューニングされた BERT を用いる．質問の生成には，既存の機械翻訳モデルを用いた．実験の結果，提案した手法がどのようなクエリにおいて性能の向上が見られるかを明らかにした．

キーワード　情報検索，質問応答，機械読解，質問生成，アドホック検索

1　はじめに

情報検索の中心的なタスクである「アドホック検索タスク」は，与えられたクエリに対して，適合度が高い順に文書を並び替える問題である．クエリからユーザの情報要求を捉え，情報要求を満たすような文書を高適合と推定する．これとは異なるタスクとして「機械読解タスク」がある。これは質問と文章が与えられた時に，文章からその質問に対する答えを抽出するタスクである．例えば,「ベトナムの首都は？」という質問と「ベトナムは東南アジアにある国で，首都はハノイです」という文章が与えられたとする。このとき、機械読解モデルは文章から「ハノイ」を答えとして抽出することが期待される．

アドホック検索タスクは長年にわたって取り組まれてきた問題である．情報検索のワークショップである Trec Web Track や NTCIR は 1990 年代から開催されており，近年では MS MARCO（a large scale MAchine Reading COmprehension dataset）[1] の Document ranking dataset や TREC Deep Learning Track [2] といったデータセットが存在している．一方で，機械読解タスクは近年特に注目を集めている [3] [4]．機械読解タスク向けのデータセットである MS MARCO や SQuAD（Stanford Question Answering Datase）[5] といった大規模なデータセットの登場により，機械読解モデルはその性能を大幅に改善し，ニューラルネットワークに基づく様々なモデルが提案されてきている．BERT を機械読解タスクにファインチューニングしたモデルでは，SQuAD において人間を超えるパフォーマンスを報告している [4].

これらの問題はそれぞれ別の問題として取り組まれ発展してきたが，ある文字列に合致する内容を取得するという面では目的が一致しており，これまでに提案されてきたモデルには多くの共通点が存在する．アドホック検索では，与えられたクエリに合致するような内容の文書を文書群から取得することを目的とし，機械読解タスクでは質問に合致するような回答を文章中から取得することを目的としている．

本論文では，この 2 つの異なる問題の共通点に着目し，一方のモデルによって他方の問題を解くことができないかという問いを立てた。特に機械読解モデルを用いることによってアドホック検索タスクを解く方法を提案する．アドホック検索タスクを機械読解モデルで解くために，我々は AIRRead（**A**d-hoc **I**nformation **R**etrieval model based on machine **Read**ing comprehension and question generation）を提案する．AIRRead では検索クエリからその背後にある質問を生成し，その質問に対する答えを含むかどうかを機械読解モデルによって判定することで，文書の適合性を推定する．一方のモデルによって他方の問題が十分な精度で達成できるのであれば，一方のモデルの発展がそのまま他方の問題への貢献へとつながり，より効率的な技術発展が期待できると考えた．本研究では機械読解モデルを用いて情報検索を行うため，機械読解モデルの性能の向上が，アドホック検索タスクでの性能の向上に直接的に繋がるようになるはずである．加えて，機械読解モデルによってアドホック検索タスクを十分な精度で解くことができれば，アドホック検索のデータセットがないような言語であっても，その言語の機械読解のモデルがあれば，アドホック検索が可能になる．

機械読解モデルとして事前学習済みの BERT を機械読解・質問応答タスクのデータセットである SQuAD2.0 でファインチューニングしたモデルを用いる．SQuAD2.0 では，機械読解タスクにおける答えの抽出だけでなく，回答可能であるかを判断する必要もある [6]．質問に対する答えが文書中に含まれている可能性が高ければその文書は高適合であるという仮定の元で，機械読解モデルに質問と文書を入力し，出力として得られる文章中のある位置から答えが始まる確率分布の最大値を適合度として利用した．検索クエリから質問を生成する手法は，文字列から文字列への変換であるため，翻訳タスクとして捉え，既存の機械翻訳モデルを用いた．機械翻訳モデルの学習には，

質問からクエリを生成することで，クエリと質問がペアになるようなデータセットを構築した．データセットの構築に使用した質問には機械読解タスクのデータセットである MS MARCO の質問を使用した．機械読解モデルによる文書のランク付けは BM25 によるランク付けされた文書リストの上位 10 件に対して行なった．

実験では提案手法のアドホック検索の性能の評価を行なった．データセットにはアドホック検索タスクである NTCIR WWW-2・WWW-3 の English タスクのデータセットを用い、アドホック検索についての性能をベースラインと提案手法を比較した．さらに，質問生成の手法をいくつか検討し，質問生成の改善がどのように機械読解モデルによる文書のランク付けに影響するかを検証した．また，機械読解モデルによる適合性推定の際に得られる答えについて分析を行なった．その結果，BM25 との比較において，機械読解モデルによる文書のランク付けを部分的に行うことで，アドホック検索の性能が改善することが判明した．

この論文における我々の貢献を以下に示す：(1) 機械読解モデルを用いてアドホック検索タスクを解くためのフレームワークを提案した．(2) 機械読解モデルとして BERT を，質問生成モデルとして注意機構のある Encoder-Decoder を使用してアドホック検索タスクを解く手法を提案した．(3) 実験を行い，提案した手法がどのようなクエリにおいてアドホック検索の性能の改善が見られるかを明らかにした．

本論文の構成は以下の通りである．2 節ではアドホック検索に関する関連研究，および機械読解に関する関連研究について述べる．3 節では問題設定を説明し，機械読解モデルのアドホック検索タスクへの適応方法について述べる．4 節では実験結果を示す．最後に，5 節では今後の課題と共に本論文の結論を述べる．

2 関 連 研 究

本節では，BERT（Bidirectional Encoder Representations from Transformers）によるアドホック検索に関する関連研究，および機械読解に関する関連研究について述べる．

2.1 アドホック検索

BERT に基づくアドホック検索では大きく分けて，BERT に文書とクエリを別々に入力することで埋め込み表現を得る手法と，BERT の出力を文書の適合度として用いる手法の 2 つに分けられる．BERT をクエリと文書の埋め込みに用いる手法では，BERT によって検索対象となる文書とクエリを別々にベクトルに埋め込む [7] [8]．BERT はクエリや文書をベクトル表現に埋め込む際に用いられ，文書の適合度の推定には cosine 類似度や内積等のベクトル間の類似度を計算する．Zhan らは，BERT によってクエリと文書を文脈化された埋め込み表現へと変換し，クエリと文書の埋め込み表現間のコサイン類似度を適合度とする手法を提案した [8]．

BERT の出力を文書の適合度として用いる手法では，文書とクエリを一度に BERT へ入力する [9] [10]．その出力の一部を，クエリに対する文書の適合度として利用することで，アドホック検索タスクを解く．Yilmaz らは，BERT に入力可能なトークンの数は制限されているため，文書を直接入力することができないという課題を解決するために，文書を文章ごとにわけた上で文書の適合度を推定する手法を提案した [9]．分割した文章とクエリの適合度を BERT によって推定し，文章ごとの適合度を統合することで最終的な文書の適合度とする．Nogueira らは BERT を 2 文書の適合性を比較する duoBERT を提案した [10]．duoBERT では，クエリと文書 2 つを 1 つのシークエンスにまとめて BERT に入力し，一方の文書がもう一方の文書より高適合である確率を抽出する．

これらの研究と同様に，BERT をアドホック検索タスクに用いる．しかし，BERT をクエリと文書の埋め込みに用いる手法と異なり，本論文では適合性の推定を BERT で行い，文書とクエリを一度に BERT に入力する．BERT の出力を文書の適合度として用いる手法とは，適合度の推定を BERT によって行う点で共通しているが，本論文では機械読解モデルとして BERT をアドホック検索に利用し，アドホック検索タスクでのファインチューニングは行わないという点で異なる．本論文では学習済みの機械読解モデルの出力から適合度を計算する．

2.2 機 械 読 解

機械読解は質問応答で用いられる技術であり，ある質問に対する答えを文章の中から抽出するタスクである．大塚らは入力された質問を，さらに詳細な内容を含む質問に変換してから機械読解モデルに入力する手法を提案した [11]．詳細な内容を含む質問に変換してから機械読解モデルに入力することで，曖昧な質問に対して回答の特定性が高まることを検証した．本論文との共通点として，機械読解モデルに入力する前に，入力する文字列をより具体性のある質問へと変換するという点が挙げられるが，本論文ではクエリから質問を生成し，本論文で扱うタスクはアドホック検索タスクである点で異なる．

質問応答において機械読解は答えの抽出を担っているが，オープンドメインな質問応答タスクでは，機械読解タスクに入力するための文章を効率的に探す必要がある．西田らはオープンドメインな機械読解タスクにおいて，文章の検索と答えの抽出を同一のモデルで行うマルチタスク学習を取り入れる手法を提案した [12]．文章の検索がアドホック検索タスクに相当し，答えの抽出が機械読解タスクに相当する．機械読解タスクとアドホック検索タスクを同一のモデルによって行うという点で本論文と類似しているが，本論文ではアドホック検索タスクに取り組み，機械読解タスクを扱わない．また，この研究では 1 つのモデルを両方のタスクで学習を行なっているが，本論文では機械読解モデルに学習済みのモデルを使用し，アドホック検索タスクに向けた学習を行わないという点で異なっている．

3 提 案 手 法

本節では，クエリを質問を生成し，機械読解モデルによって

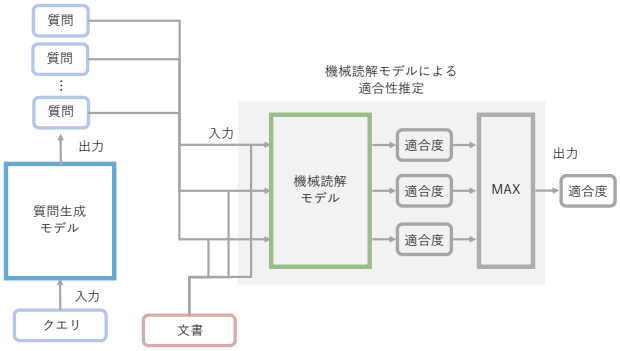

図 1　提案するフレームワークの概要図

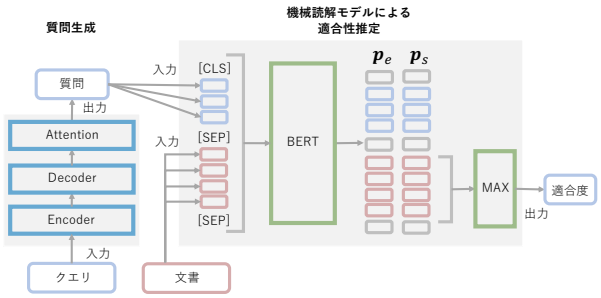

図 2　提案手法の構成

適合性推定を行う問題について説明を行う．

3.1　問題設定

本論文は，D を文書集合とし，与えられたクエリ q_r に基づいて，D に含まれる文書 d_i の適合度 s_i を推定し，適合度に基づいて文書をランク付けする問題を解く．ただし，本論文では適合度の推定に機械読解モデルを使用し，上位 k 件の文書のランキング $(d_1, d_2, \ldots, d_k)$ を返す．機械読解モデルには学習済みの機械読解モデルを使用し，アドホック検索向けの学習は行わない．

3.2　フレームワーク

本論文で提案する機械読解モデルを用いてアドホック検索タスクを解くためのフレームワークを図 1 に示す．クエリが与えられた時に，文書集合 D から BM25 のような索引付けによって高速に検索可能な検索モデルで適合性推定を行い，D の部分集合 D_{BM25} を得る．クエリ q_r は機械読解モデルへの入力のために質問 q_s へと変換し，D_{BM25} に対して機械読解モデル f による適合性の推定を行う．機械読解モデルには質問と文書を入力し，適合度 s_i を得る．質問は質問生成モデル g によってクエリをベースに生成される．機械読解モデル及び質問生成モデルによって i 番目の文書とクエリの適合度を推定する場合，以下のように適合度を推定する．

$$q_s = g(q_r)$$
$$s_i = f(q_s, d_i)$$

上記のように推定した適合度に基づいて，文書のランク付けを行う．

クエリを質問に変換する際，本来の情報要求を汲むためにクエリと情報要求を共有している必要がある点に注意する．また，クエリの情報要求が曖昧である場合，複数の質問を生成することも考えられる．例えば,「和歌山　観光」というクエリが与えられてたとき，考えられる情報要求として「和歌山県での観光地を知りたい」や「和歌山の観光課のページを探してる」「和歌山大学の観光学部について知りたい」等の多様な情報要求が考えられる．その場合，複数の質問について文書の適合度を推定し，複数の適合度を一つに集約し，最終的な適合度とする．クエリから複数の質問を生成し，その最大値を適合度とした場合，以下のように行う．

$$Q_s = g(q_r)$$
$$s_i = \max_{q \in Q_s} f(q, d_i)$$

ただし，Q_s は生成された質問の集合である．$g(q_r)$ でクエリから質問の集合を生成し，生成した質問それぞれについて得られた適合度の最大値 $\max_{q \in Q_s} f(q, d_i)$ をクエリ q_r に対する文書 d_i の適合度とする．得られた適合度に基づいて文書をランキングすることで最終的なランキングを作成する．

Algorithm 1 Generate a query from a question

```
tokens ← Tokenize(question)
length ← GetLength()
length ← max(length, Length(question))
query ← {}
for all i ← 1 ... length do
    token ← arg max_{token∈tokens} IDF(token)
    query ← query ∪ {token}
end for

return query
```

3.3　AIRRead

提案手法である AIRRead について説明する．図 2 に提案手法の大まかな流れを示す．

3.4　質問生成

本論文では，学習済みの機械読解モデルを使うため，クエリから質問を生成する．クエリから質問を生成する手法はいくつか提案されており，クエリログから質問テンプレートを生成する手法や，LSTM によるテキスト変換で実現する手法がある [13] [14] [15]．本論文では，クエリから質問を生成する過程をテキストからテキストへの変換する過程と捉え，機械翻訳タスクとして扱う [15]．そのため，機械翻訳モデルを用いてクエリから質問へと翻訳することで，質問の生成を実現する．機械翻訳モデルには，注意機構を利用した Encoder-Decoder を用いる [16]．機械翻訳モデルを学習させるために，クエリと質問のペアデータが必要となる．本論文では質問からクエリを生成

することでクエリと質問がペアになるようなデータセットを構築した．質問からクエリを生成する手順を Algorithm 1 に示す．なお，トークン t の idf は以下のように計算した:

$$idf(t) = \log \frac{|\text{Questions}|}{|\{q_s : t \in \text{tokens}\}|}$$

ただし，|Questions| はデータセットに含まれる総質問数であり，tokens は質問 q_s をトークンに分割した集合である．質問からクエリを生成するために，クエリを最初のステップでは質問をトークンに分割する（Tokenize(question)）．クエリは一般に質問より短く，Web クエリの長さの平均は 2.4 であり，3 単語以下のクエリが約 76%を占める [17]．そのため，生成するクエリの長さを 1 から 3 の範囲で一様分布にランダムに決定する（GetLength()）．このとき，生成するクエリの長さが質問を超えないよう注意する．生成するクエリは質問の情報要求を共有している必要があるため，質問中に含む語からクエリとして利用する語を抽出し，抽出する語は，単語の出現頻度が低いほどその単語がより多くの情報量を持っているという仮定の元で，その語の IDF の大きさによって決定する．その後，決定したクエリ長の回数だけ，質問のトークンの最も IDF 値が高いものを 1 つ取り出し（$\underset{token \in tokens}{\arg\max}$ IDF(token)），クエリの集合に追加する（query $\leftarrow$ query $\cup$ {token}）．

3.5 機械読解による適合性推定

本節ではアドホック検索における適合度の推定を，学習済みの機械読解モデルによって行う手法を提案する．

一般に，機械読解モデルは質問応答タスクに用いられ，文書と質問を入力として受け取り，与えられた文章中から質問に対する答えを抽出する [3]．答えは，文章中における区間として示され，機械読解モデルからは文章の各トークンについて，そのトークンから答えの区間が始まる確率，及びそのトークンで答えが終わる確率が出力される．

本論文では BERT を機械読解モデルとして用いる．BERT による機械読解モデルでは，質問と文章を繋げて 1 つのシークエンスとして入力し，入力された各トークンについて答えの区間が始まる確率と終わる確率を出力する．質問 q_s と文章 a を入力するとき，質問と文章の間と入力文末に [SEP] を追加し，入力文頭に [CLS] を追加め，質問の長さを $\text{len}(\text{q}_\text{s})$，文章 len(a) とすると，入力シークエンスの長さは $l = \text{len}(\text{q}_\text{s}) + \text{len}(\text{a}) + 3$ となる．この場合，出力は l の長さの確率分布が二つ得られる．一方がそのトークンから答えの区間が始まる確率 $\mathbf{p_s} = (p_{s,1}, p_{s,2}, \ldots, p_{s,l})(\in \mathbb{R}^l)$，もう一方がそのトークンで答えの区間が終わる確率 $\mathbf{p_e} = (p_{e,1}, p_{e,2}, \ldots, p_{e,l})(\in \mathbb{R}^l)$ である．

本論文では，質問に対する答えがあると判断できる文章は高適合であるという仮定の元で，得られた $\mathbf{p_s}$ の文章に対応する確率の最大値を適合度として計算する．文章中のトークンに対応した $\mathbf{p_s}$ を $\mathbf{p_a} = (\mathbf{p_{s,j}}, \mathbf{p_{s,j+1}}, \ldots, \mathbf{p_{s,j+\text{len(a)}}})(j = \text{len}(\text{q}_\text{s}) + 2)$ とすると，q_s における文章 a の適合度 $s_{q_s,a}$ は次のように計算される．

表 1　構築した質問生成データセットの概要.

	データ数	平均クエリ長	平均質問長
クエリ-質問データセット	727,858	1.99	6.38

表 2　実際のデータの一部.

クエリ	質問
stalin eastern why	why did stalin want control of eastern europe
nails rusty	why do nails get rusty
depona ab	depona ab
is world	is the atlanta airport the busiest in the world

$$s_{q_s,a} = \max_{1 \leq i \leq \text{len(a)}} p_{a,i}$$

本論文では BERT を SQuAD2.0 でファインチューニングする [6]．SQuAD2.0 は機械読解タスクのデータセットであるが，SQuAD1.0 と異なり文章から回答不可能な質問も含まれている．そのため，質問が回答不可能であると判断された場合は，入力シークエンスの先頭である [CLS] トークンの位置が答えの区間となるように BERT を学習する．これは回答不可能である可能性が高い場合は [CLS] の位置から答えの区間が始まると推測される確率が高まり，相対的に他の位置から答えの区間が始まると推測される確率が低くなることを意味する．既に説明したように，答えは文章中のトークンに対応する確率から計算するため，答えがないと機械読解モデルによって判断できる場合は，計算される適合度も低くなると考えられる．

アドホック検索タスクでは与えられたクエリについて文書の適合度を計算する必要があるが，BERT による機械読解モデルでは入力可能なシークエンスの長さに上限があるため，クエリと文書からできる入力シークエンスの長さが上限を超える場合は，文書の適合度を直接推定できない．この問題を解決するために，入力シークエンスの長さが上限を超えないように文書を文章に分割し，各文章と質問を機械読解モデルに入力することで適合度を推定する．この時，一つの質問-文書ペアに対して複数個の適合度が生まれるため，本論文では各文章の適合度の最大値を，そのクエリにおけるその文書の適合度として用いる．質問 q_s と，文書 d_i を文章集合 A に分割して適合性を推定する際，d_i の適合度 $s_{q_s,i}$ は次のようになる．

$$s_{q_s,i} = \max_{a \in A} f(q_s, a)$$

本論文では，$s_{q_s,i}$ を s_i として用いる．

4 実　　験

本節ではまず使用するデータセットについて説明する．質問生成モデルの学習ではデータセットの構築を行なったため，データセットの作成と統計情報について説明する．それからベースライン手法について述べ最後に実験結果を示す．

4.1 データセット

クエリから質問へ変換するモデルを学習するために，MS MARCO の質問を用いて (クエリ-質問) ペアとなるようなデータセットの構築を行なった．構築手順は 3.4 で説明した通りで

ある．構築したデータセットの統計情報を表 1 に，実際のデータの一部を表 2 に示す．表 2 にあるように，質問長が 3 以下の質問の場合は，生成されたクエリが質問と一致することもある．

提案手法の評価には，データセットとして NTCIR WWW-2 と WWW-3 を用いた．

4.2 実験設定

我々は実験におけるベースライン手法として，NTCIR WWW-2・WWW-3 でベースラインとして提供されている BM25(www) と提案手法で利用する BM25 でのランキング BM25(our)，NTCIR15 WWW-3 English SubTask において最も良い性能を達成した手法 である Birch（以下 Birch とする）を用いた [18] [19]．Birch は BERT によるアドホック検索モデルであり，文書を文章ごとにわけた上で文書の適合度を推定することにより，文書の適合度を推定する手法である [9]．評価指標には nDCG@10・Q@10・nERR@10 の 3 つを用いる．Q は sakai により提案されたランク付けされた文書のリストを評価する指標で，平均精度に累積利得を取り入れ，多値適合性で利用可能な形へと拡張した評価指標である [20]．nERR は ERR を正規化した指標である．ERR は Reciprocal rank に，ランキングの上から文書を見ていき，文書の適合度が高いほどユーザはその文書を見て満足しやすく，検索から離脱しやすいという cascade ユーザモデルを導入した評価指標である [21]．機械読解モデルには，事前学習された BERT を SQuAD2.0 でファインチューニングしたモデルを用いた．質問生成モデルには注意機構を利用した Encoder-Decoder を用い，4.1 で説明した質問とクエリがペアになっているデータセットで学習した．学習時の損失関数にはバイナリ交差エントロピーを用いた．複数個の質問を生成し適合性推定を行なったが，性能の改善への貢献が見られなかったため，質問を生成する際にはクエリから質問を 1 つ生成した．機械読解モデルによる適合度の推定は Batch 数 16 で行なった．

AIRRead のバリエーションとして，以下の設定のものを利用する．

Partially　固有名詞であるトークンを含んだクエリに対してのみ AIRRead を適応する手法．固有名詞であるトークンを含まないクエリについては，BM25 での文書のランキングをそのまま出力する．AIRRead を適応する条件については，詳細を 4.3.1 に示す．

Handmade　人の手により質問が作られた場合の性能を測るために，NTCIR15 WWW-3 と WWW-2 の English subtask で与えられたクエリを手動で質問へと変換し，その質問を用いて適合性推定を行った手法．変換は筆者によって行い，クエリの情報要求が説明されている description フィールドを読み，クエリを質問へと変換した．これにより，質問生成モデルの性能に依存しない形で学習済みの機械読解モデルでのアドホック検索の能力を測ることができると考えられる

表 3　各ランキング手法の上位 10 件での実験結果．

手法	WWW-3			WWW-2		
	nDCG	Q	nERR	nDCG	Q	nERR
BM25(www)	0.575	0.585	0.676	0.326	**0.304**	0.478
BM25(our)	0.628	0.639	0.744	0.317	0.291	0.459
Birch	**0.694**	**0.712**	**0.796**	**0.334**	0.300	**0.486**
AIRRead	0.627	0.636	0.735	0.303	0.281	0.424

WhatIs　機械読解モデルに入力する質問を，質問生成モデルによって作成する代わりに，与えられたクエリの文頭に「What is」をつけることで質問とする手法．例として,「blue note」がクエリとして与えられた場合，質問は「What is blue note」となる．

Copy　質問生成モデルである注意機構付きの Encoder-Decoder に，コピー機構を適応した手法．コピー機構は，入力されるシークエンスの一部を出力する文字列に移植する機構であり，本論文では入力となるクエリに含まれるトークンが，生成する質問でも出力されやすくなる．これにより，入力する文字列中に未知語がある場合でも，出力することが可能となる．

BM25 機械読解モデルによるスコア s_{AIRRead} と BM25(our) によるスコア $s_{BM25(\mathrm{our})}$ を次のように組み合わせた手法．

$$s = \alpha \cdot s_{\mathrm{AIRRead}} + (1-\alpha) \cdot s_{BM25(\mathrm{our})} \quad (1)$$

WWW-3 でハイパーパラメータである α のチューニングを行い，$\alpha = 0.005$ と決定した．

4.3 実験結果

ベースライン手法と我々が提案する手法の実験の結果を表 3 に示す．WWW-2，WWW-3 両方のデータセットにおいて，AIRRead は比較した全ての評価指標で BM25(www) より良い結果を出したが，Birch を下回る結果となった．AIRRead が BM25(www) を上回る結果となったが，BM25(our) の結果を見ると，全ての評価指標において AIRRead より BM25(our) の方がより良い結果となっている．このことから，AIRRead では BM25 で得られたランキングを機械読解モデルによって並び替えることで最終的な文書のランキングを出力しているため，AIRRead での機械読解モデルによるアドホック検索の性能の改善は得られなかったと考えられる．

4.2 で説明した AIRRead の各バリエーションにおける WWW-2・WWW-3 での実験結果を表 4 に示す．AIRRead(Handmade) は，WWW-3 では全ての評価指標において AIRRead を下回る結果となったが，WWW-2 では全ての評価指標において AIRRead を上回った．AIRRead(WhatIs) の結果と比較すると，WWW-3 では Q と nERR で AIRRead(WhatIs) を下回っており，nDCG では同等の結果である．WWW-2 でも同様に Q を除く nDCG と nERR で AIRRead(WhatIs) を上

表 4　AIRRead の各バリエーションでの実験結果．* はそのデータセット上でファインチューニングした時の数値であることを示している．

手法	WWW-3			WWW-2		
	nDCG	Q	nERR	nDCG	Q	nERR
AIRRead	0.627	0.636	0.735	0.303	0.281	0.424
AIRRead(Handmade)	0.621	0.631	0.719	0.309	0.284	0.447
AIRRead(Partially)	0.627*	0.635*	0.745*	**0.320**	**0.295**	**0.474**
AIRRead(WhatIs)	0.621	0.632	0.713	0.308	0.287	0.436
AIRRead(Copy)	0.618	0.628	0.709	0.305	0.282	0.423
AIRRead(BM25)	**0.631***	**0.644***	**0.751***	0.318	0.291	0.466
AIRRead(Handmade かつ Par tially)	0.629*	0.639*	0.749*	0.319	**0.295**	0.472
AIRRead(WhatIs かつ Partially)	0.627*	0.635*	0.737*	0.319	**0.295**	0.469
AIRRead(BM25 かつ Partially)	0.629*	0.640*	0.744*	0.318	0.292	0.463

表 5　著者により作成された質問における疑問詞の割合．

疑問詞	WWW-3	WWW-2
what	0.875	0.684
where	0.063	0.127
who	0.025	0.0633
when	0.013	0.0
how	0.013	0.038
which	0.013	0.089

回っている．しかし，WWW-2 と WWW-3 の両方で，各指標についての AIRRead(WhatIs) と AIRRead(Handmade) の差は小さい．表 5 に著者により作成された質問のデータセットごとの疑問詞の割合を示す．この表から，WWW-2 と WWW-3 の両方で疑問詞が what である質問が最も多く，次点との差は WWW-3 では 0.875，WWW-2 では 0.684 と大きいため，作成された質問の大半が what から始まっている質問であることがわかる．このうち What is から始まる質問は，WWW-2 で 87%，WWW-3 で 70%であった．このことから，AIRRead(WhatIs) と AIRRead(Handmade) の評価の差が小さいことは，適合性の推定に用いた質問が類似している部分が大きかったことに起因している可能性がある．

AIRRead(BM25) は，WWW-2 と WWW-3 の両方で AIRRead を全ての評価指標において上回っている．この結果は，機械読解モデルによる適合性推定で得たスコアに，BM25 でのスコアを取り入れることが AIRRead のアドホック検索の性能の改善に貢献していることを示している．BM25 の上位 10 件のリランキングを行うことが，BM25 により推定された適合性を考慮していると考えることができるが，明示的にスコアに組み込むことで，機械読解モデルで行うことのできない単語ベースのマッチングの情報を文書のリランキングにおいて反映でき，より高い性能を示していると思われる．

AIRRead(Handmade かつ Partially) は固有名詞であるトークンを含んだクエリに対してのみ AIRRead(Handmade) を適応した手法であり，AIRRead(WhatIs かつ Partially) は固有名詞であるトークンを含んだクエリに対してのみ AIRRead(WhatIs) を適応した手法である．AIRRead(Handmade かつ Partially) と AIRRead(WhatIs かつ Partially) を比較すると，WWW-2 と WWW-3 の両方において，全ての評価指標で AIRRead(Handmade かつ Partially) が AIRRead(WhatIs かつ Partially) を上回るか，同等となる結果となっている．この結果から，固有名詞におけるリランキングでは質問の改善がランキングの結果に貢献する可能性があると思われる．

BM25(our) での文書のランキングから，機械読解モデルによるリランキングでどれほど改善するかをクエリごとに調べる．機械読解モデルによるリランキングでの改善率を次のように定義する．

$$\text{改善率} = \frac{\text{MSnDCG}_{\text{RC}}}{\text{MSnDCG}_{\text{BM25}}}$$

ただし，MSnDCG$_{RC}$ は機械読解モデルによる文書のランキングの MSnDCG，MSnDCG$_{BM25}$ は MSnDCG$_{\text{our}}$ による文書のランキングの MSnDCG とする．MSnDCG$_{BM25}$ が 0 の時は，MSnDCG$_{RC}$ も 0 となるため，改善率は 0 とする．

AIRRead(Copy) は WWW-3 において全ての評価指標で AIRRead を下回っているが，WWW-2 においては nDCG と Q において AIRRead を上回っている．この結果から，コピー機構により出力される質問中の未知語の改善は，アドホック検索での性能の改善に貢献しなかったことがわかる．質問生成において未知語となりやすい語として，クエリ中に含まれる固有名詞が挙げられるが，第一段階として行われる BM25 によるランク付けによって，間接的にクエリ中の固有名詞の情報を利用していることが起因していると思われる．

4.3.1　AIRRead の部分適応

AIRRead(Partially) で行なった AIRRead を部分的に適応する手法について説明する．

WWW-3 のクエリについて，AIRRead(Handmade) のランキングを改善率で降順にソートし，上位 20 件と下位 20 件のクエリに含まれる品詞の割合を調べた．表 6 に，全体と上位 20 件，下位 20 件での品詞の割合と，上位 20 件と下位 20 件の各品詞の割合の差で降順にソートした結果を示す．固有名詞の改善率の上位 20 件と下位 20 件の差が最も大きくなっていることから，固有名詞を含むクエリについては AIRRead によるリランキングが効果的であるという考察が可能である．名詞は上位 20 件と下位 20 での割合の差が最も小さくなっているが，WWW-3 でのクエリ全体での名詞の割合が 0.459 と高いこと

表 6　改善率でクエリを降順にソートした時の，上位 20 件のクエリ中に含まれている品詞の割合．差は上位 20 件でのその品詞の割合から，下位 20 件でのその品詞の割合を引いた値．

品詞	全体	上位 20 件	下位 20 件	上位 20 件 - 下位 20 件
固有名詞	0.238	0.486	0.245	+0.241
副詞	0.022	0.057	0.019	+0.038
その他	0.006	0.029	0.000	+0.029
動詞	0.066	0.057	0.038	+0.019
形容詞	0.077	0.086	0.094	-0.008
助詞	0.028	0.000	0.019	-0.019
接続詞	0.006	0.000	0.019	-0.019
限定詞	0.022	0.000	0.019	-0.019
設置詞	0.050	0.029	0.075	-0.046
名詞	0.459	0.257	0.472	-0.215

表 7　WWW-3 のクエリと AIRRead の改善率．

クエリ	改善率	固有名詞を含む
kangaroo	1.498	✓
george washington university	1.440	✓
scorpions	1.339	
Pirates of the Caribbean	1.317	✓
zeus	1.285	
Texas Hold'em	0.841	✓
akron beacon journal	0.784	✓
Smart home	0.686	
Movies about animals	0.657	
internet pros and cons	0.565	

を考慮すると，名詞を含むようなクエリで AIRRead によるリランキングが性能の改善に負の影響を及ぼしていると結論づけるには更なる調査が必要であると思われる．

WWW-3 での実験では，改善率上位に固有名詞を含むクエリの割合が相対的に多いことを確認した．表 4 の AIRRead(Partially) および AIRRead(Handmade かつ Partially) によると，固有名詞による AIRRead の部分適応の有効性は WWW-2 上でも確認できる．WWW-2 において，AIRRead と AIRRead(Partially) を比較すると，AIRRead(Partially) は全ての評価指標において AIRRead を上回っている．同様に，AIRRead(Handmade かつ Partially) は WWW-2 において全ての評価指標において AIRRead(Handmade) を上回っている．表 3 にある WWW-2 での BM25(our) との比較においても，AIRRead(Partially) は BM25(our) を全ての評価指標において上回っている．このことは，部分適応によって BM25(our) による文書のランキングから，クエリに基づいて選択的にリランキングを行うことで，機械読解モデルによるリランキングが最終的なアドホック検索での性能の向上に貢献したこと示している．

表 7 に WWW-3 での AIRRead の改善率を降順にソートしたリストの上位 5 件と下位 5 件について，クエリと改善率を示す．「固有名詞を含む」の列には，クエリに固有名詞が含む場合はその行に ✓ が入る．この表から，上位 5 件中に固有名詞を含まないクエリがあり，下位 5 件中に固有名詞を含むクエリがあることがわかる．改善率上位のクエリには固有名詞を含まないものも存在しているため，これらのクエリが与えられた際に機械読解モデルによるリランキングを行えるようにクエリを分類できれば，さらなる性能の改善が期待できる．同様に改善率下位 5 件には固有名詞を含むものも存在していることは，これらのクエリに対して BM25 によるランキングをそのまま出力することで，機械読解モデルによる文書のランクでの負の影響を抑え，AIRRead の性能の改善が期待できることを示している．

4.3.2　機械読解モデルによる答え

表 8 に，AIRRead(Handmade) によるランキングで上位 1 位の適合性推定に利用された質問と，その時の機械読解モデルにより得られた適合度と答え，及びその時の改善率を示す．クエリ「country music awards」が与えられたときの WWW-3 での検索意図は「You want to know what music awards there are for country music.」である．機械読解モデルによって得られた答えは「cmt music awards」であり，これは CMT によるカントリーミュージックの授賞式を指しており，質問に対する答えとして適切であると思われる．クエリ「greeting cards」では，検索意図は「Where can I find a free e-mails greetting cards You want to send an email greeting card to your friend, so you would like to find some free ones on the Internet.」であり，Transactional なクエリであることがわかる．質問では「What site should I use for sending greeting cards」であり，Web サイトを答えとして期待しているが，答えとして pathy が得られた．この時の文章を含む文書のラベルは 1 であり，機械読解モデルによる適合度は高く推定されているが，得られた答えは適切とは言えない．この結果は，機械読解モデルが高適合と判断し，実際に適合であった文書であっても，期待される答えが得られないケースもあることを示しており，クエリと情報要求を十分に共有している質問ではなかった可能性が考えられる．クエリ「kangaroo」が与えられた際は，答えとして male が得られた．これは質問の回答として適切とは言えないが，改善率は 1 を上回っており，文書のランク付けには成功している．また，上位 5 件までのランキングで得られた答えの中で，質問に対する回答として適合している答えは存在せず，精度の高いランキングが得られた場合でも適切な答えを抽出できない場合があることがわかる．

5　ま　と　め

本論文では，クエリから質問を生成し，学習済みの機械読解モデルを利用してアドホック検索タスクを解く問題に取り組んだ．この問題を解くためのフレームワークを提案し，提案手法として質問生成モデルには注意機構のある Encoder-Decoder を利用し，機械読解モデルには BERT を利用した．実験では NTCIR WWW-3，WWW-2 を用いてベースライン手法と提案手法の比較を行なった．その結果，BM25 との比較において，固有名詞を含むクエリについて学習済みの機械読解モデルは文書のリランキングで有効に作用することが判明した．今後の課題としては，固有名詞を含むクエリでも機械読解モデルによるリランキングが負の影響を及ぼすこともあったことを背景に，

表 8　AIRRead(Handmade) で使用された質問と，そのときのランキング上位 1 位の改善率と機械読解モデルから得られた適合度及び答え．

クエリ	質問	答え	適合度	改善率
kangaroo	What different types of kangaroos there are	male	0.991	1.509
country music awards	What music awards are there for country music	cmt music awards	0.995	1.097
greeting cards	What site should I use for sending greeting cards	pathy	0.999	0.966

有効に作用するクエリの詳細な傾向を明らかにすることが考えられる．

謝辞　本研究は JSPS 科研費 21H03554，21H03775 の助成を受けたものです．ここに記して謝意を表します．

文　献

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[2] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M. Voorhees, and Ian Soboroff. Trec deep learning track: Reusable test collections in the large data regime. 2021.

[3] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[4] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[7] Omar Khattab and Matei Zaharia. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*, page 39–48. Association for Computing Machinery, New York, NY, USA, 2020.

[8] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. RepBERT: Contextualized Text Embeddings for First-Stage Retrieval, 2020.

[9] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3490–3496. Association for Computational Linguistics, 2019.

[10] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert, 2019.

[11] 大塚淳史, 西田京介, 斉藤いつみ, 浅野久子, 富田準二, and 佐藤哲司. 質問意図の明確化に着目した機械読解による質問応答手法の提案. volume 34, pages A–J14_1–12, 2019.

[12] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. ACM, Oct 2018.

[13] Gideon Dror, Yoelle Maarek, Avihai Mejer, and Idan Szpektor. From query to question in one click: Suggesting synthetic questions to searchers. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 391–402, New York, NY, USA, 2013. Association for Computing Machinery.

[14] Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. Automatically generating questions from queries for community-based question answering. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 929–937, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[15] Adarsh Kumar, Sandipan Dandapat, and Sushil Chordia. Translating web search queries into natural language questions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[16] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[17] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. Searching the web: The public and their queries. volume 52, pages 226–234, 2001.

[18] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, and Ian Soboroff. Overview of the ntcir-15 we want web with centre (www-3) task. 01 2021.

[19] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, et al. Overview of the ntcir-15 we want web with centre (www-3) task. *Proceedings of NTCIR-15. to appear*, 2020.

[20] Tetsuya Sakai. New performance metrics based on multigrade relevance: Their application to question answering. 2004.

[21] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. CIKM '09, page 621–630, New York, NY, USA, 2009. Association for Computing Machinery.