情報処理学会論文誌　データベース　Vol.14　No.4　36–48 (Oct. 2021)

被検索文書の絞り込みと補強，クエリ拡張に基づく統計データ向けアドホック検索

岡本 卓[1,a)]　宮森 恒[1,b)]

受付日 2021年3月9日, 採録日 2021年7月2日

概要：本稿では，被検索文書の絞り込みと補強，クエリ拡張に基づく統計データに対するアドホック検索手法を提案する．近年，政府や様々な団体が保有する公共的データを日常生活や社会のために有効活用するためのオープンデータの利用基盤整備が世界的に進んでおり，オープンデータの一種である統計データに対するアドホック検索基盤の重要性が高まっている．統計データは一般に表形式で記載されており，文章形式で記載される従来のアドホック検索の被検索文書とは異なる特徴を持つ．本稿では，被検索文書とクエリをカテゴリ分類し，候補となる被検索文書を絞り込む手法，統計データのメタデータにはない情報を統計表本体から抽出し，被検索文書を補強する手法，および，クエリに類似した拡張語を用いる手法で構成されるランキング手法を提案する．実験では，提案手法の構成要素の様々な組合せで性能を比較し，最良となる組合せを検証する．

キーワード：統計データ検索，テキスト分類，カテゴリ，表理解，ヘッダ抽出，クエリ拡張

Ad Hoc Search for Statistical Data Based on Refinement and Augmentation of Retrieved Documents and Query Expansion

Taku Okamoto[1,a)]　Hisashi Miyamori[1,b)]

Received: March 9, 2021, Accepted: July 2, 2021

Abstract: In this paper, we propose an ad hoc search method for statistical data based on narrowing down and augmenting documents to be searched and query expansion. In recent years, there has been a worldwide trend towards the development of an open data infrastructure for the effective use of public data held by governments and other organisations for the benefit of everyday life and society, and the importance of ad hoc search infrastructures for statistical data, typically used as open data, is increasing. Statistical data is generally described in tabular format, and has different characteristics from the documents to be searched by conventional ad hoc search, which are described mailnly in text format. In this paper, we propose a ranking method which consists of three parts: (1) a method to categorize retrieved documents and queries to narrow down candidate documents, (2) a method to augment retrieved documents by extracting information from the statistical table itself which is not in the metadata of the statistical data, and (3) a method to use extended words which are similar to queries. In the experiments, we compare the performance of various combinations of the components of the proposed method and verify the best combination.

Keywords: statistical data retrieval, text classification, categories, table comprehension, header extraction, query expansion

1. はじめに

近年，政府や様々な団体が保有する公共的データを日常生活や社会のために有効活用するためのオープンデータの利用基盤整備が世界的に進んでおり，オープンデータの一

[1] 京都産業大学大学院先端情報学研究科
Division of Frontier Informatics, Kyoto Sangyo University, Kyoto 603–8555, Japan
[a)] i2086042@cc.kyoto-su.ac.jp
[b)] miya@cc.kyoto-su.ac.jp

種である統計データに対するアドホック検索基盤の重要性が高まっている．

たとえば，統計データは，社会問題となっているフェイクニュースに対処するための事実確認（ファクトチェック）において重要な役割を果たすと考えられる．一般に，政府統計や業界団体などが公開している統計データは一定の品質が担保されていると考えられるが，それと比較照合することで情報の事実性の不備を確認できるためである．

一方，これまでの研究では，文書中の表検索や表理解に関する研究は活発に行われてきたものの，統計データそのものを対象としたアドホック検索についてはこれまでほとんど扱われてこなかった．

2020 年に開催された評価型ワークショップ NTCIR-15 では，Data Search タスクとして，統計文書を対象としたアドホック検索タスク [1] が実施された．そこでは，統計データに対するアドホック検索の基盤技術の確立を目的として，政府統計ポータルサイト（e-Stat）で提供される日本の統計データを対象としたアドホック検索の課題が提起され，本課題の重要性が注目されつつある．

本研究では，このタスクと同じ設定の課題に取り組む．すなわち，統計データに対するアドホック検索の基盤技術の確立を目的として，政府統計ポータルサイト（e-Stat）で提供される日本の統計データを対象としたアドホック検索の課題に取り組み，本課題に対する解決策を提案する．

扱う対象文書は，政府統計データから抽出したメタデータ，統計データ本体から構成されており，メタデータは文書長が短く，統計データ本体もタイトルなどを除くとほとんどの場合，数値で構成されており，表構造も統一されていないという特徴がある．この形式の対象文書を本稿では統計文書と呼ぶこととする．

統計文書は検索の手がかりとなる情報が少ないため，我々はユーザクエリが意図する問題領域の範囲を適切にとらえることを目的として，統計文書をカテゴリで絞り込むカテゴリ検索を提案する．また，メタデータの文書長の短さを補うため，統計データ本体から表のヘッダ情報を抽出し，被検索文書の一部とするデータ補強手法，および，クエリ拡張を組み合わせた手法を提案する．実験では，統計文書群に対する検索結果のクエリとの関連性を，NTCIR-15 Data Search タスクで採用された評価方法に準拠しながら，クラウドソーシングと，クラウドソーシングと同等の評価をした著者以外の研究室学生を利用して評価し，提案手法を構成する要素の様々な組合せでの有用性を検証する．

本稿の構成は以下のとおりである．2 章で関連研究について述べ，3 章で本稿で使用する統計文書データセットと既存研究の文書との違いについて述べる．4 章で本稿で扱う問題を定式化し，5 章で提案手法について詳述する．6 章で実験方法と実験結果を示し，7 章で考察を述べる．最後に 8 章で結論と課題を述べる．

2. 関連研究

情報検索は古くから研究されてきた分野であり，クエリに関連する文書を検索するため，各文書に様々なスコアを付与する手法がいくつも提案されてきた．主な手法としては，文書やクエリをベクトルに変換して類似度などを計算するベクトル空間モデル [2] や，クエリと文書の関連する確率を計算する確率モデル [3]，ニューラルネットを使用した推論ネットワークモデル [4] などがあげられる．特に確率モデルの 1 つである BM25 [5] は，現在でも有用性が高く，様々な検索エンジンにおいて広く利用されている．近年では，推論ネットワークモデルを発展させた深層学習を利用した手法 [6] や自然言語理解の分野で目覚ましい進展をもたらしている言語モデル BERT [7] を使用したランキング手法 [8] も提案されている．

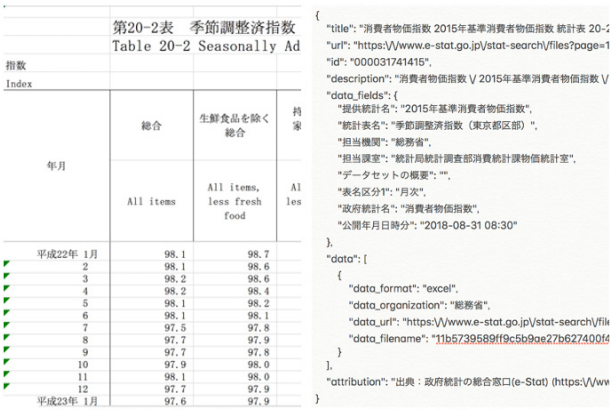
表を対象とした検索についても，これまでに多くの研究がなされている．Zhang らは，WikiTables コーパスに基づく表検索のためのデータセットを作成し，クエリと表を複数の意味空間で表現し，それらを様々な類似性尺度でマッチングする手法を提案している [9]．Shraga らは，表の内的類似尺度と表の外的尺度を用いてリランキングする手法を提案している [10]．Chen らは，表から選択された項目と，クエリ，関連フィールドを BERT を用いて学習する手法を提案している [11]．これらの研究は，いずれも WikiTables データセットを用いているが，このデータセットは，Wikipedia 文書の表を用いているため，表のサイズが比較的小さく，表本体のセル値が数値でない通常の単語である割合が比較的大きいという特徴がある．一方，本研究で扱うデータセットは，e-Stat から収集された NTCIR-15 Data Search のデータセットを用いており，表のサイズが比較的大きく，表本体のセル値が数値である割合が大きいという特徴がある点でこれらの従来研究と異なる．

統計データを対象とした検索の研究としては，文章が参照する統計データを検索する手法を中野ら [12] が提案している．統計データは本研究と同じ e-Stat から抽出したものを使用しているが，クエリがキーワードではなく，文章である点で本研究とは異なる．

NTCIR-15 では，統計文書を対象としたアドホック検索タスク [1] が実施された．本研究が扱う問題は，このタスクと同じ設定である．NTCIR-15 内で提案された手法の多くで事前学習モデル BERT が利用されているが，現在までのところ，従来の通常文書を対象としたアドホック検索でのランキング性能を大きく超える結果には至っていない．

3. データセット

本稿では，被検索文書集合として，NTCIR-15 Data Search 日本語タスクで提供されたデータセットを使用する（以後，統計文書データセットと呼ぶ）．統計文書データセットの



(a) 統計データの例　　(b) メタデータの例

図 1　統計文書の例

Fig. 1　Example of statistical document.

表 1　統計データのファイル形式の分布

Table 1　Distribution of file formats of statistical data.

ファイル形式	頻度
xls	686,436
csv	568,042
pdf	49,124
xlsx	34,794
xlsm	6

1 文書は，**図 1** に示すように，1 つの統計データと，その書誌情報を記載した 1 つのメタデータの組で構成されており，各文書は，日本の政府統計ポータルサイト（e-Stat）から収集されている．

統計データは，xls や csv などの形式で保存された統計データ本体のファイルに相当する．統計データは，1 つ以上の表形式のデータを含み，表のタイトルや表のヘッダには数値以外の通常のテキストが記載されているが，表本体は，ほとんどの場合，非常に多くの数値で構成されている．**表 1** に，統計データのファイル形式の分布を示す．

メタデータは，e-Stat の統計データの導入ページに記載されているデータを抽出したものである．メタデータは JSON 形式のファイルであり，統計データの id，統計データの導入ページの URL，統計データの title のほか，統計データの簡潔な概要を記述した description，統計データ本体の URL，ファイル形式，ファイル名などを表す変数名と，それぞれに対応する値で構成されている．

表 2 に，統計データにおける数値と数値以外の単語の使用割合の平均と標準偏差，および，統計データの総数を示す．ここで，単語の使用割合は，統計データに含まれる全文字列を MeCab を用いて単語に分割し，1 単語を構成する文字がすべて半角数字である場合に数値を表す単語，それ以外の場合に数値以外の単語と判定した．表 2 より，統計データでは，数値を表す語が平均約 79%の割合で使用されていることが分かる．

同様に，**表 3** に，メタデータにおける数値と数値以外の単語の使用割合の平均と標準偏差，および，メタデータの総数を示す．メタデータでは，数値以外の語が平均約 90%の割合で使用されており，統計データに比べて，検索の手がかりとしやすい，数値以外の単語の割合が大きいことが分かる．

表 2　統計データにおける数値と数値以外の単語の使用割合

Table 2　Raio of use of numeric and non-numeric words in statistical data.

数値		数値以外の文字		統計データの総数
平均	標準偏差	平均	標準偏差	
0.786	0.221	0.213	0.221	1,338,402

表 3　メタデータにおける数値と数値以外の単語の使用割合

Table 3　Ratio of use of numeric and non-numeric words in metadata.

数値		数値以外の文字		メタデータの総数
平均	標準偏差	平均	標準偏差	
0.076	0.023	0.923	0.023	1,338,402

表 4　メタデータで使用される単語長

Table 4　Word length used in metadata.

title		description		title and description	
平均	標準偏差	平均	標準偏差	平均	標準偏差
36.2	16.7	20.0	8.2	56.2	21.2

次に，メタデータの文書長について説明する，図 1 で示すとおり，メタデータにおいては，title，description 変数以外の変数に対応する値は，単なる id や URL など，必ずしも検索の手がかりとしやすい内容とはなっていない．そこで本稿では，メタデータの title，description 変数の値で構成される組を，メタデータの文書と見なすこととする．**表 4** にメタデータの title，description 変数のそれぞれの値に含まれる単語数の平均と標準偏差，および，メタデータの総数を示す．ここで，単語数は，当該文字列を MeCab で分割した結果の単語数を表す．

アドホック検索では，従来，Web 文書，学術論文，議論フォーラム，政府や企業の内部文書，ニュースやソーシャルメディアの記事など，様々な文書が被検索文書として用いられてきたが，これらの文書は，主に日常的に用いられる文章などの自然言語で記述されている点が特徴である．たとえば，小説などの比較的長い文書長を持つ文書では約 3 万単語 [13] が用いられ，比較的短い文書長の新聞記事では約 330 単語 [14] が用いられている．

一方，本稿で対象とするメタデータで検索の手がかりとしやすい自然言語で記述された変数 title，description の単語数の平均は 56 単語程度と非常に少ない．

そのため，メタデータに数値以外の割合が多くても単語数が少ないため，従来のアドホック検索手法をそのまま適用するだけでは，クエリを適切に満たす検索結果を得るの

は難しい．

4. 問題の定式化

本章では，本稿で扱う問題を定式化する．まず，クエリ集合 Q，被検索文書集合 D をそれぞれ

$$Q = \{q_i\}, \quad D = \{d_j\} \tag{1}$$

とする．ここで，クエリ q_i は 1 回の検索で与えられる 1 つ以上の単語列 $w_1^{q_i}, w_2^{q_i}, \ldots, w_{n_{q_i}}^{q_i}$ を表し，1 つの被検索文書 d_j は，1 つのメタデータ m_j，および，1 つの統計データ t_j の組として表される．

$$d_j = (m_j, t_j) \tag{2}$$

被検索文書集合 D のうち，クエリ q_i に関連がある文書集合は

$$D^{q_i,+} = \{d_j^{q_i,+}\} \tag{3}$$

と表すこととする．

また，クエリ q_i に関連がある文書集合 $D^{q_i,+}$ を，ランキング関数 $rank(q_i, d_j^{q_i,+})$ で降順にソートしたランキングリストを $R_{rank(q_i,d_j^{q_i,+})}$ で表すこととする．

以後，集合 $S = \{s_a\}$ のうち，クエリ q_i に関連がある要素の集合を $S^{q_i,+} = \{s_a^{q_i,+}\}$ で表すこととする．

本稿で取り組む問題の目的は，統計文書集合 D から，クエリ q_i に関連がある文書集合の適切なランキング結果 $result_{q_i}$ を取得することである．すなわち，

$$result_{q_i} = R_{rank(q_i,d_j^{q_i,+})} \tag{4}$$

とすることである．

5. 提案手法

本章で 3 つの提案手法について述べる．3.1 節でカテゴリ検索手法，3.2 節でデータ補強，3.3 節でクエリ拡張についてそれぞれ説明する．

5.1 カテゴリ検索

統計文書に対する検索のために，ユーザのクエリが意図する検索範囲を適切にとらえて被検索文書集合をカテゴリで絞り込む手法を提案する．インデキシング時には，テキスト分類器を用いて各被検索文書にカテゴリを付与し，カテゴリ付きの新たな被検索文書集合として登録する．検索時には，クエリからテキスト分類器を用いてカテゴリを推定し，推定されたカテゴリに属する被検索文書集合に対してのみランキングを実行し，検索結果を返す．カテゴリ検索の処理手順を**図 2** に示す．

以下，処理手順を詳しく説明する．まず，カテゴリ集合 C を次のように定義する．

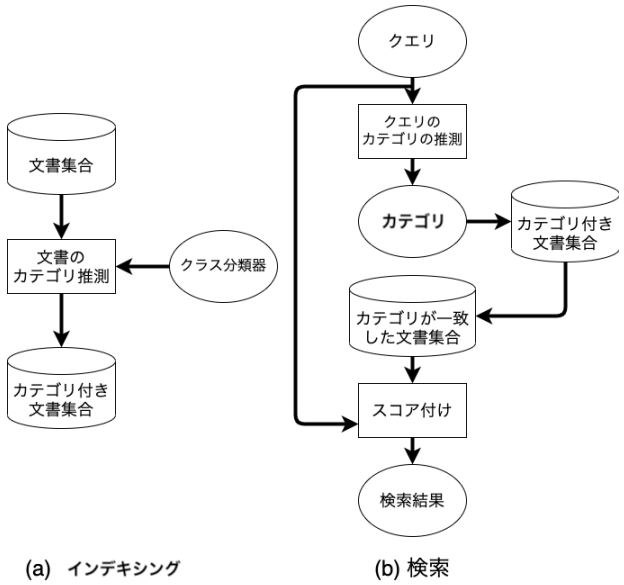


図 2　カテゴリ検索の概要
Fig. 2　Overview of category search.

$$C = \{c_p\} \tag{5}$$

ここで，c_p は事前に定義された 1 つのカテゴリラベルを表す．

カテゴリ集合 C は，以下の手順で定める．まず，コミュニティ質問応答 Web サービス Yahoo! 知恵袋のサイト検索で，“e-Stat” を意味する計 9 種類の表記の異なるクエリ[*1]をそれぞれ用いて得られる検索結果をすべて収集する．検索結果 1 件は，1 つの質問とそれに対応する 1 つ以上の回答の組から構成されている（以後，これを 1 件の質問・回答アイテムと呼ぶ）．クエリは，質問と回答のいずれか，あるいは，その両方に含まれる可能性が考えられるため，収集した質問・回答アイテムから，回答に e-Stat へのリンクが含まれているものを抽出し，その各々に対応する質問が属するカテゴリを列挙した．以上により，Yahoo! 知恵袋の大分類として用いられている 17 カテゴリから**表 5** に示す 10 カテゴリを C として定めた（2020 年 4 月 20 日時点）．各カテゴリの質問・回答アイテムの件数は，平均 149.4，標準偏差 6.3 である．

次に，カテゴリを推定するテキスト分類器を構築する．収集した質問・回答アイテム集合の各アイテムについて，名詞と動詞の品詞を持つ単語を抽出し，該当する単語の fastText による分散表現の平均を各アイテムの特徴ベクトルとした．この特徴ベクトルと正解カテゴリを用いて SVM で学習することでテキスト分類器を構築した．SVM のカーネルは linear とし，$C = [1, 1000]$ の範囲でグリッドサーチを行い，$C = 1$ とした．構築したテキスト分類器の

[*1] 本稿では，“e-stat”，“E-stat”，“e-Stat”，“E-STAT”，“estat”，“Estat”，“eStat”，“ESTAT”，“イースタット” の 9 種類を用いた．

表 5　カテゴリ検索で使用したカテゴリと収集した質問・回答アイテム数

Table 5　Categories used in category search and number of QA items collected.

カテゴリ	# of QA items
インターネット	145
エンタメ	156
テクノロジ	140
デバイス	143
マナー	151
健康	151
子育て	149
親子関係	150
知恵袋	146
生活	163

分類精度は 0.69 となった[*2].

以下，作成したテキスト分類器を関数 $text\_clf(x) \in C$ で表し，引数 x として，被検索文書 d_j やクエリ q_i を受け付けるものとする．

カテゴリ検索のインデキシング時には，被検索文書 d_j だけでなく，カテゴリラベル $text\_clf(d_j)$ が付与された状態で索引づけされるものとする．

検索時には，クエリ q_i に対して，カテゴリラベル $text\_clf(q_i)$ を推定し，そのカテゴリラベルでフィルタリングされた文書集合 $D^{c_{q_i}}$ を取得する．$D^{c_{q_i}}$ のうち，クエリ q_i に関連がある文書集合 $D^{c_{q_i},q_i,+}$ に対して，ランキング関数でソートしたランキング結果 $result_{q_i}$ を取得する．すなわち，

$$D^{c_{q_i}} = \{d_j^{c_{q_i}}\} = \{d_j | text\_clf(q_i) = text\_clf(d_j)\} \quad (6)$$

$$result_{q_i} = R_{rank(q_i, d_j^{c_{q_i},q_i,+})} \quad (7)$$

となるように検索が実行される．

ただし，検索結果 $result_{q_i}$ の件数 $|result_{q_i}|$ が閾値 θ に対して，$|result_{q_i}| < \theta$ となる場合は，カテゴリ検索の結果だけでなく，カテゴリ検索を適用しない場合の検索結果も用いる．具体的には，元の被検索文書集合 D に対して，ランキング結果を取得し，それをカテゴリ検索のランキング結果の下位に連結する．

$$result_{q_i} = R_{rank(q_i, d_j^{c_{q_i},q_i,+})} || R_{rank(q_i, d_j^{q_i,+})} \quad (8)$$

ここで，|| はリストの結合を表す演算子とする．

5.2　データ補強

メタデータの文書長の短さを補うため，統計データ本体から表のヘッダ情報を抽出し，被検索文書に追加して扱う手法を提案する．

[*2] 本文中の質問・回答アイテムの特徴ベクトルの構成方法や学習手法，カーネル種別などについては，予備実験において，他の様々な組合せと比較した結果，最も高い分類精度を示したものである．

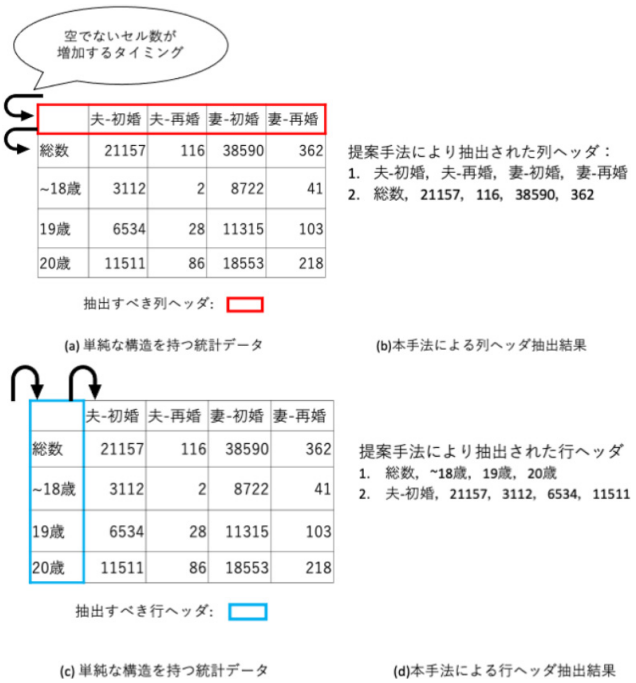


図 3　単純な構造を持つ統計データと本手法によるヘッダ抽出結果

Fig. 3　Statistical data with simple structure and header extraction results by our method.

具体的には，統計データ内の各行あるいは各列における空でないセル数を，行あるいは列ごとの順に調べ，空でないセル数の変化によって列あるいは行ヘッダをそれぞれ抽出する．統計データは，図 **3** (a)/(c)，図 **4** (a)/(c) にそれぞれ示すとおり，単純な構造と階層的な構造を持つものに大別でき，本手法はいずれの場合においても有効に機能することを目指したものである．

たとえば，列ヘッダについて考える．

単純な構造を持つ場合も階層的な構造を持つ場合も，基本的に，本体に含まれる横方向のセル数と同数のセル数を持つ列ヘッダが少なくとも 1 つは存在することが考えられる．図 3 (a) の場合，本体 1 行目の (‘21557’, ‘116’, ‘38590’, ‘362’) のセル数と列ヘッダ (‘夫–初婚’, ‘夫–再婚’, ‘妻–初婚’, ‘妻–再婚’) のセル数はいずれも 4 で同数である．また，図 4 (a) の場合も，本体 1 行目の (‘757’, ‘157’, ‘428’, ‘24’) のセル数と列ヘッダ 3 行目の (‘15～24 歳’, ‘25～34 歳’, ‘雇われてする仕事’, ‘自営業主’) のセル数はいずれも 4 で同数である．

また，階層的な構造を持つ場合は，列ヘッダ内の上位階層の行から下位階層の行に向かって，空でないセル数は増加する傾向が見られる．図 4 (a) の場合，列ヘッダの 1 行目から 3 行目に向かって空でないセル数が，1，2，4 と増加している．

以上のことから，本手法では，列ヘッダを抽出する際，各行ごとのセル数の変化に着目することとした．同様の議論は，行ヘッダを抽出する際にも成立する．

列ヘッダを抽出する場合の抽出手順を Algorithm 1 に示

図 4　複雑な構造を持つ統計データと本手法によるヘッダ抽出結果
Fig. 4　Statistical data with complex structure and header extraction results by our method.

Algorithm 1 Extracting column headers from statistical data

Input: statistical data sd
Output: column headers hdr_col

```
prev = 0
hdr_col = []
max_row = sd.rows.length
for i = 1, ..., max_row do
  curr = sd.rows[i].unempty_cells().length
  if curr > prev then
    hdr_col.append(sd.rows[i].unempty_cells())
  end if
  prev = curr
end for
return hdr_col
```

す．まず，入力された統計データを sd とし，直前の行の空でないセル数を格納する変数 $prev$ を 0 で，列ヘッダを格納する変数 hdr_col を空のリストでそれぞれ初期化する．統計データの各行の並びをリストとして格納した $sd.rows$ の要素数を max_row に格納する．

次に，1 行目から max_row 行まで以下を繰り返す．セルの並びが格納されているリストから，空でないセルのみをフィルタリングして抽出し，その結果をリストとして返すメソッドを $unempty\_cells()$ とする．第 i 行目に含まれるセルのリスト $sd.rows[i]$ に対して，$unempty\_cells()$ を適用し，第 i 行目に含まれる空でないセルのリストを取得する．第 i 行目の空でないセルのリストの要素数 $length$ を，第 i 行目の空でないセル数として $curr$ に格納する．第 $i-1$ 行目の空でないセル数は $prev$ に格納されている．

$curr$ が $prev$ よりも大きければ，リストの最後に別のリストを追加する $append()$ メソッドを用いて，列ヘッダ hdr_col の最後に，第 i 行目の空でないセルのリストを追加する．$prev$ を $curr$ で更新し，次の行の処理に移行する．

繰返しが終了したら，列ヘッダ hdr_col を h_j^{col} に返す．h_j^{col} には，列ヘッダに該当する，空でないセルを格納した行のリストが格納されている．行ヘッダについては，Algorithm 1 の行と列を相互に入れ替えた内容の処理を実行し，ヘッダ h_j^{row} を抽出する．抽出したヘッダ情報をメタデータ m_j に追加することで，メタデータの文書長の短さを補った文書 d_j^{m+h} を作成する．

本手法は，セルに対する操作が可能な xls，xlsx，xlsm，csv 形式のファイルに適用する．しかし，セルへの直接操作が困難な pdf 形式のファイルについては，ファイル内の全文字列を抽出した結果を m_j に追加し，d_j^{m+h} とする．

図 3，図 4 に，単純な構造，複雑な構造をそれぞれ持つ統計データに対して，本手法を適用した結果を示す．図中の (a)，(c) が列ヘッダを抽出する場合，(b)，(d) が行ヘッダを抽出する場合を表す．ここでは，列ヘッダの抽出を例に説明を進める．図 3 (b) の列ヘッダの抽出結果を見ると，正しいヘッダ (‘夫–初婚’, ‘夫–再婚’, ‘妻–初婚’, ‘妻–再婚’) が抽出できていることが分かる．一方，その次の行である (‘総数’, ‘21557’, ‘116’, ‘38590’, ‘362’) も抽出結果に含まれているが，この 2 つ目の結果は，適切な抽出結果とはいえない（以後，問題点 1 と呼ぶ）．図 4 (b) の列ヘッダの抽出結果を見ると，正しいヘッダ (‘労働力調査’), (‘年齢階級’, ‘希望している仕事’), (‘15～24 歳’, ‘25～34 歳’, ‘雇われてする仕事’, ‘自営業主’) が抽出できているが，その次の行である (‘男女計’, ‘757’, ‘157’, ‘428’, ‘24’) も抽出結果に含まれていることが分かる（問題点 1）．

複雑な構造を持つ統計データにおいては，**図 5** (a) に示すように，列ヘッダ内の上位階層の行から下位階層の行に向かって，空でないセル数が同じ行が連続する場合もある（図中の (‘対教師暴力’) と (‘全国’)）．この場合，本手法による列ヘッダの抽出結果は，図 5 (b) に示すとおり，(‘対教師暴力’), (‘区分’, ‘学校総数’, ‘発生学校数’, ‘発生件数’) となり，(‘全国’) の検出漏れが起きてしまう（以後，問題点 2 と呼ぶ）．問題点 2 の改善策として，列ヘッダの最初の検出位置から最後の検出位置までのすべての行を抽出する方法が考えられる（以後，検出開始終了位置法と呼ぶこととする）．

問題点 1，2 が存在しながら，本手法で特段の改善策を

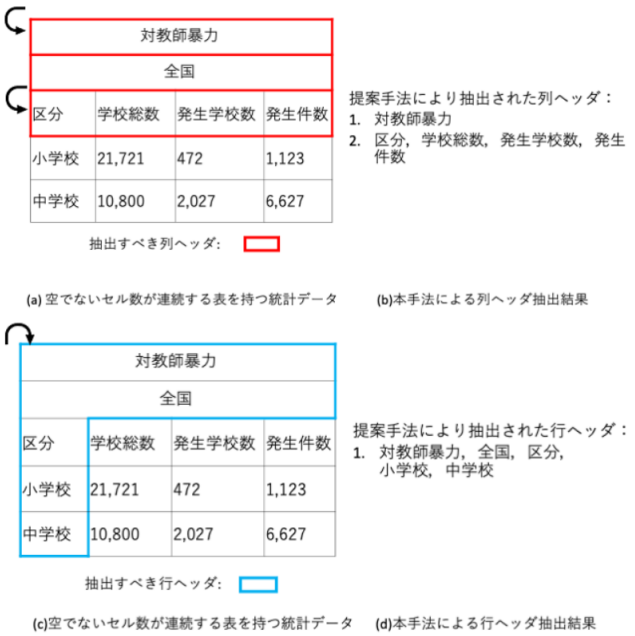


図 5　複数の票を持つ統計データと抽出結果

Fig. 5　Statistical data with multiple votes and extraction results.

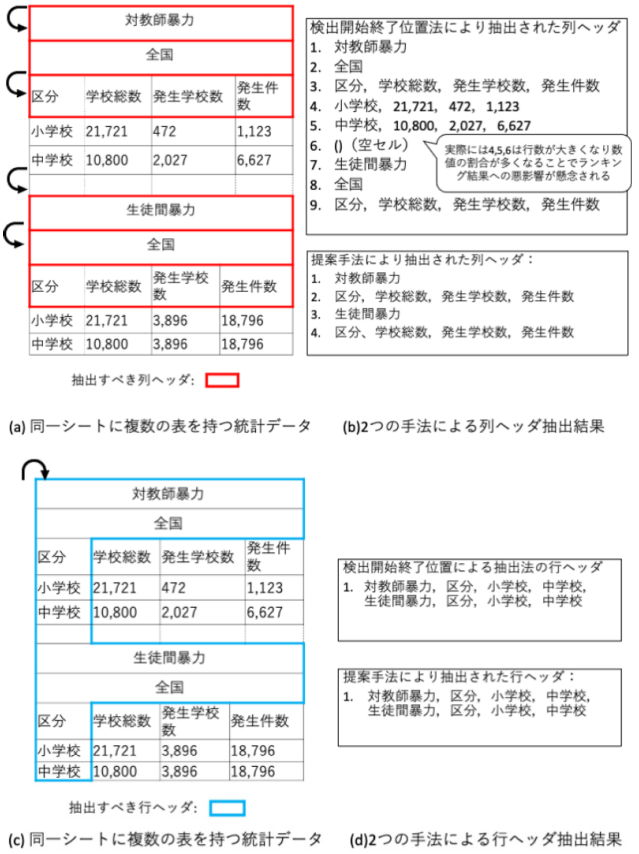


図 6　複数の表を含む統計データと 2 つの手法によるヘッダ抽出結果

Fig. 6　Statistical data including multiple tables and header extraction results by two methods.

講じなかったのは，**図 6** (a)/(c) に示すような，1 つの統計データ内に複数の表が存在する場合に対処するためである．

図 6 (a) に対して，検出開始終了位置法を適用して列ヘッダを抽出すると，図 6 (b) に示すように，(‘対教師暴力’) から，(‘生徒間暴力’) の次の行の (‘区分’, ‘学校総数’, ‘発生学校数’, ‘発生件数’) までがすべて抽出されてしまう（以後，問題点 3 と呼ぶ）．ここで，‘小学校’，‘中学校’ でそれぞれ始まる 2 行は，行ヘッダ以外が数値のみで構成されているが，実際には，このような行が非常に多く抽出され，数値が占める割合が大きくなるため，ランキング結果への悪影響が懸念される．

これに対し，提案手法を適用して列ヘッダを抽出すると，図 6 (b) に示すように，(‘対教師暴力’)，(‘区分’, ‘学校総数’, ‘発生学校数’, ‘発生件数’)，(‘生徒間暴力’)，(‘区分’, ‘学校総数’, ‘発生学校数’, ‘発生件数’) となり，行ヘッダ以外が数値のみで構成される行の検出数を適切に抑制することができる．

以上のように，提案手法では，問題点 1，2，3 を比較したときに，問題点 3 の影響が最も大きいと考え，それを回避することを重視した．

本手法が，どの程度ヘッダを正しく抽出できているかを検証するために定量評価を行った．まず，統計文書データセットからランダムに統計文書を 100 件抽出し，各文書に対して，列ヘッダに該当する行の並びを人手で同定し，その並びの各行内の全文字列を 1 件の正解データとして記録する．行ヘッダについても同様に正解データを記録する．次に，各文書から本手法でヘッダ抽出した結果を評価する．本手法で抽出した列ヘッダのある 1 行の中に含まれる文字列が，列ヘッダのある行の正解データとして記録された文字列と完全一致した場合にのみ正解と判定し，これ以外の場合は不正解と判定する．行ヘッダについても同様の基準で判定する．

抽出された 100 件のファイル形式の内訳は，xls が 54 件，csv は 44 件，pdf が 1 件，xlsx が 1 件であった．ヘッダ情報の抽出手法の定量評価結果を**表 6** に示す．表 6 より，本手法は，正しいヘッダ情報をかなり網羅的に抽出できる手法であることが確認できる．

表 6　提案手法によるヘッダ抽出の評価結果

Table 6　Evaluation results of header extraction by the proposed method.

ヘッダ	適合率	再現率	F 値
行ヘッダ	0.545 (236/433)	0.975 (236/242)	0.699
列ヘッダ	0.489 (523/1068)	0.906 (523/577)	0.635
{ 行, 列 } ヘッダ	0.505 (759/1501)	0.927 (759/819)	0.654

5.3　クエリ拡張

与えられたクエリのみでは，被検索文書との的確なマッチングが行われず，不十分な検索結果しか得られない場合が考えられる．そこで，クエリの各単語と類似した単語を補強するクエリ拡張を採用することで，より適切な統計

表 7　質問データセットにおけるカテゴリと質問数
Table 7　Question types and number of questions in the question dataset.

カテゴリ	質問数
健康，美容とファッション	446,427
子育てと学校	227,992
ビジネス，経済とお金	125,273
教養と学問，サイエンス	471,616
その他	256,368
スポーツ，アウトドア，車	418,029
生き方と恋愛，人間関係の悩み	754,768
エンターテイメントと趣味	997,955
暮らしと生活ガイド	401,358
地域，旅行，お出かけ	219,568
スマートデバイス，PC, 家電	325,036
ニュース，政治，国際情勢	192,096
インターネット，通信	207,050
Yahoo! JAPNAN	80,057
マナー，冠婚葬祭	53,796
コンピュータテクノロジー	28,627

データを取得する手法を提案する．

クエリの各単語と類似した単語を取得するために，QA サイトの質問を情報要求と考え，QA サイトで用いられる質問を収集したデータセットを作成した（以後，質問データセットと呼ぶ）．質問データセットは，Yahoo! 知恵袋データ（第 3 版; 2019，2020 年度提供版）[*3]からカテゴリと質問本文を抽出することで作成した．

こうして集めたデータセットの数を**表 7** に示す．

質問データセットを用いて Skip-gram モデルを訓練し，クエリを単語ベクトルに変換する．Skip-gram の訓練には，質問データセットの名詞，動詞，形容詞，形容動詞のいずれかの品詞の単語のみを使用する．これは，クエリとして使われる語は基本的に体言あるいは用言に該当する語であると考えられるためである．クエリ拡張をする際は，クエリ q_i の各単語 $w_k^{q_i}$ について，対応する単語ベクトルのコサイン類似度が最大となる 1 単語 $w_k^{'q_i}$ を取得し，拡張語としてクエリに追加する．これにより，クエリ q_i の 2 倍の語数に拡張された拡張クエリ q_i' が以下のとおり得られる．

$$q_i' = \{w_1^{q_i}, w_2^{q_i}, \ldots, w_{n_{q_i}}^{q_i}, w_1^{'q_i}, w_2^{'q_i}, \ldots, w_{n_{q_i}}^{'q_i}\} \quad (9)$$

検索時には，クエリ q_i に対して，拡張クエリ q_i' を用いて検索を実行し，ランキング結果 $result_{q_i}$ を取得する．

$$result_{q_i} = R_{rank(q_i', d_j^{q_i', +})} \quad (10)$$

6. 評価実験

本章では，提案手法の様々な組合せで，統計文書データセットに対するランク付けされたリストを生成し，従来のアドホック検索と同様の方法で各手法の有用性を評価する．

6.1 実験方法

評価に用いるクエリとして，NTCIR-15 Data Search 日本語タスクで提供されたテストクエリ 96 件を使用する [1]．

実験での評価方法について説明する．評価方法は NTCIR-15 Data Search タスクで採用された評価方法に準拠する．はじめに，各テストクエリから得られた検索結果のうち上位 10 件の文書を取得する．評価者には，上位 10 件のうちの 1 文書と，クエリの作成元となった質問文のペアについて，L0，L1，L2 の 3 段階で関連性を評価する作業を指示した（L0：関連がない，L1：部分的に関連がある，L2：関連がある）．作業データとして既知データを加えることで，質の悪い評価者を除外し，品質を保つようにしている．評価には，クラウドソーシングを利用した．以上が NTCIR-15 Data Search タスクで採用された評価方法であるが，本実験においては，クラウドソーシングと，クラウドソーシングと同等の評価をした著者以外の研究室学生を利用した．クラウドソーシングによる評価では，NTCIR-15 Data Search タスクで行われた評価条件と同一の条件を指定することで，評価結果の比較に問題がないことを担保している．

ここでは，6.2 節で示す 3 つの比較手法，および，提案手法 RUN-1，3，4 について関連性を評価した．本実験で関連性を評価する上位 10 件の文書としては，NTCIR-15 Data Search タスクで評価された文書を含めてすべての文書を評価した．NTCIR-15 Data Search タスクで評価された文書には，クラウドソーシングにより 3 段階の関連性ラベルが付与されており，その結果はタスク参加者の中で共有されている（著者らもタスク参加者である）．本実験において特定の文書の評価結果が NTCIR-15 Data Search での評価結果と異なる場合は，NTCIR-15 Data Search での評価結果を優先した．これにより，NTCIR-15 Data Search での評価結果との一貫性を維持し，かつ，NTCIR-15 Data Search で未評価の文書についても漏れなく関連性を評価した上で手法間の比較をすることができる．

研究室学生による評価では，1 名のワーカが実施可能な作業タスク数以外の条件は，NTCIR-15 Data Search タスクで行われた評価条件と同一の条件を指定した（1 名のワーカが実施可能な作業タスク数について，クラウドソーシングでは 100 としたのに対し，研究室学生では 1,056 とした）．ここでは，6.2 節で示す提案手法 RUN-2 について関連性を評価した[*4]．

[*3] ヤフー株式会社（2019，2020）：Yahoo! 知恵袋データ（第 3 版）．国立情報学研究所情報学研究データリポジトリ（データセット）．https://doi.org/10.32130/idr.1.3

[*4] RUN-2 のみ研究室学生による評価を実施したのは，学内事情により 2021 年 3 月にクラウドソーシングによる評価を実施できなかったためである．なお，後日 RUN-2 に対しても RUN-1 などと同じ条件でクラウドソーシングによる評価を実施したところ，研究室学生による評価結果と同様の傾向が見られ，論文中での議論に影響しないことを確認している．

表 8　評価対象とする手法の一覧
Table 8　Combination of evaluation methods.

手法名	NTCIR-15 Data Search タスクにおける RUN-ID	BM25	PRF	BERT	QM	カテゴリ検索	データ補強	クエリ拡張
Baseline		✓						
BM25+PRF+BERT [15]	NII-J-EX-10	✓	✓	✓				
BM25+QM [16]	uhai-J-10	✓			✓			
RUN-1	KSU-J-EX-3	✓				✓		
RUN-2		✓					✓	
RUN-3	KSU-J-EX-1	✓				✓	✓	
RUN-4		✓				✓	✓	✓

PRF，BERT，QM は，それぞれ疑似適合性フィードバック，BERT による再ランキング，検索における各クエリ語の貢献度に応じたクエリ修正を表す．

6.2　評価対象とする手法

表 8 に，実験で評価対象とする手法の一覧を示す．提案手法として，カテゴリ検索，データ補強，クエリ拡張の有無で組み合わせた 4 つの手法 RUN-1，2，3，4 を評価する．また，比較手法として，BM25 によるランキング手法（Baseline と表す），パラメータ最適化した BM25 と擬似適合性フィードバック，BERT による再ランキングを組み合わせた手法（BM25+PRF+BERT と表す）[15]，検索における各クエリ語の貢献度に応じたクエリ修正と BM25 によるランキングを組み合わせた手法（BM25+QM と表す）[16] の 3 つの手法を評価する．

RUN-1 は，メタデータの title および description 変数の値で構成される m_j からなる被検索文書集合 D_m に対し，カテゴリ検索と BM25 によるランキングが適用された手法である．

RUN-2 は，メタデータの title および description 変数の値で構成される m_j に，行と列のヘッダ抽出結果 h_j^{row} と h_j^{col} が追加された被検索文書集合 D^{m+h} の関連文書に対して，BM25 によるランキングが適用された手法である．

$$D^{m+h} = \{d_j^{m+h}\} \tag{11}$$

$$result_{q_i} = R_{rank_{BM25}(q_i, d_j^{m+h,q_i,+})} \tag{12}$$

RUN-3 は，メタデータの title および description 変数の値で構成される m_j に，行と列のヘッダ抽出結果 h_j^{row} と h_j^{col} が追加された被検索文書集合 D^{m+h} に対して，テキスト分類器を用いたカテゴリ検索と，BM25 によるランキングが適用された手法である．

$$\begin{aligned} D^{m+h,c_{q_i}} &= \{d_j^{m+h,c_{q_i}}\} \\ &= \{d_j^{m+h} | text\_clf(q_i) = text\_clf(d_j^{m+h})\} \end{aligned} \tag{13}$$

$$result_{q_i} = R_{rank_{BM25}(q_i, d_j^{m+h,c_{q_i},q_i,+})} \tag{14}$$

RUN-4 は，カテゴリ分類の精度が高く，検索結果に L2 と評価される文書が比較的多いと予備実験で判断された RUN-3 の内容に，クエリ拡張を追加した手法である．

$$result_{q_i} = R_{rank_{BM25}(q_i', d_j^{m+h,c_{q_i'},q_i',+})} \tag{15}$$

表 9　ランキング結果の評価結果
Table 9　Evaluation result of ranking results.

Method	NTCIR-15 Data Search タスクにおける RUN-ID	nDCG@10
BM25		0.351
BM25+PRF+BERT [15]	NII-J-EX-10	0.406
BM25+QM [16]	uhai-J-10	0.392
RUN-1	KSU-J-EX-3	0.353
RUN-2		0.567
RUN-3	KSU-J-EX-1	0.426
RUN-4		0.534

6.3　実験結果

表 9 に，ランキング結果の評価結果を示す．カテゴリ検索と BM25 を組み合わせた手法である RUN-1 では，ベースライン手法に比べて nDCG@10 の値が 0.002 増加した．カテゴリ検索による改善がわずかに見られた．被検索文書集合を，統計データの行ヘッダと列ヘッダで補強した文書集合を用いた RUN-2 による nDCG@10 の値は，ベースラインによる値と比較して，0.181 増加した．ヘッダ部分を抽出したデータ補強により，より広く関連文書を取得でき改善につながったと考えられる．

RUN-1 での被検索文書集合を，統計データの行ヘッダと列ヘッダで補強した文書集合とした RUN-3 による nDCG@10 の値は，RUN-1 による値と比較して，0.013 増加した．カテゴリ検索と統計データの行ヘッダと列ヘッダによるデータ補強は，より広く関連文書を取得したランキングにつながると考えられる．しかし，RUN-2 による nDCG@10 の値と比べ，RUN-3 の nDCG@10 の値は 0.141 減少している．カテゴリ検索による絞り込みが悪影響を与えている可能性がある．

RUN-3 にクエリ拡張を追加した RUN-4 による

nDCG@10 の値は，RUN-3 と比べて 0.107 増加した．クエリ拡張により，統計文書のより適切なランキング結果が得られると考えられる．

最後に，比較手法 BM25+PRF+BERT，BM25+QM と提案手法を比較する．BERT による再ランキングやクエリ修正を用いた比較手法よりも，BM25 にデータ補強を組み合わせた手法で nDCG@10 の値を上回っていることが確認できる．統計文書データに対する検索では，BERT を用いた再ランキングでも十分な結果を出すのが困難である一方，単純だが表ヘッダを用いてデータ補強を行うことで，再現率が向上した可能性が考えられる．

7. 考察

本章では，手法の組合せによる nDCG@10 の値の改善と悪化がみられた組合せについて考察する．

カテゴリ検索を使用した RUN-1 の nDCG@10 の値は，ベースライン手法と比べて 0.002 上昇した．検索結果を確認すると，カテゴリに属する文書内容が比較的明確に異なる “スポーツ” と “生活” といったカテゴリでは，それらを正しく判別し絞り込みに成功している．しかし，“生活” と “健康” といったカテゴリでは，カテゴリに属する文書内容が比較的類似しており，誤判別してしまう事例が存在した．実際，“23 歳 睡眠時間” というクエリは，“生活” というカテゴリに分類されるが，関連性があるとされる医療関連の内容の被検索文書には，“健康” のカテゴリに分類されている．そのため，カテゴリ検索のみではクエリに合致した適切な検索結果を返すことが困難である場合もあることが確認された．

データ補強を使用した RUN-2 の nDCG@10 の値は，ベースライン手法と比べて 0.216 上昇した．検索結果を確認すると，クエリに関連する文書がより上位にランキングされていること，および，ベースライン手法では取得されていなかった文書が取得できていることを確認した．データ補強は統計データの検索に有用であることが確認できた．

データ補強のみを使用した RUN-2 にカテゴリ検索を加えた RUN-3 は RUN-2 の nDCG@10 の値に比べて 0.141 減少した．カテゴリ検索による絞り込みがうまく機能せず，関連性のある文書の順位が変動したことが考えられる．**表 10** に RUN-2 から RUN-3 へ変化させた際の検索結果の変化を示す．

表 10 の結果から，上位 10 件内で L1，L2 の順位が上昇した文書数は 46 件であるが，上位 10 件内で L1，L2 の順位が順位が下降した文書数は 96 件となっている．また，上位 10 件に新規に含まれた L1，L2 文書数と，上位 10 件から除外された L1，L2 文書数はともに 93 件であった．全体として，関連性のある文書の順位が下降したことにより nDCG@10 の値が減少したと考えられる．

データ補強のみを使用した RUN-2 と，それにカテゴリ検索とクエリ拡張を加えた RUN-4 の nDCG@10 を比較すると 0.034 減少した．関連性に応じた文書順位の変化を調べる．**表 11** に RUN-2 から RUN-4 へ変化させた際の検索結果の変化を示す．

表 10　RUN-2 から RUN-3 に変化させた際の上位 10 件の検索結果の変化
Table 10　Changes in the top 10 search results when changing from RUN-2 to RUN-3.

検索結果の変化の種別	L0	L1	L2	総数
上位 10 件内で順位が上昇した文書数	23	27	19	69
上位 10 件内で順位が下降した文書数	82	77	19	178
順位に変化がない文書数	208	186	98	492
上位 10 件に新規に含まれた文書数	128	56	37	221
上位 10 件から除外された文書数	128	64	29	221

表 11　RUN-2 から RUN-4 に変化させた際の上位 10 件の検索結果の変化
Table 11　Changes in the top 10 search results when changing from RUN-2 to RUN-4.

検索結果の変化の種別	L0	L1	L2	総数
上位 10 件内で順位が上昇した文書数	17	27	9	53
上位 10 件内で順位が下降した文書数	34	33	20	87
順位に変化がない文書数	265	167	64	496
上位 10 件に新規に含まれた文書数	341	82	73	496
上位 10 件から除外された文書数	265	167	64	496

表 11 の結果から，上位 10 件内で L1，L2 の順位が上昇した文書数は 36 件であるが，上位 10 件内で L1，L2 の順位が下降した文書数は 53 件となっている．また，上位 10 件に新規に含まれた L1，L2 文書数が 155 件であるのに対し，上位 10 件から除外された L1，L2 文書数は 231 件であった．全体として，関連性のある文書数が減少したことにより nDCG@10 の値が減少したと考えられる．

一方，検索結果中の L2 の文書数に着目して考察する．**図 7** に各手法による検索結果の L0〜L2 の文書数を示す．図 7 の RUN-2 の L2 の文書数に対し，RUN-3，RUN-4 の L2 の文書数を確認すると，それぞれ 8 件，9 件増加していることが分かる．すなわち，RUN-2 と比べて RUN-3，4 では，関連性が高い文書をより多く上位にランキングできている．このことから RUN-3，4 は nDCG@10 の値としては低下しているものの，カテゴリ検索を追加したことにより関連性が高い文書を上位に取得しやすいといえる．以上の結果から，データ補強のみを使用した RUN-2 が nDCG@10 で最も良好な値を示したものの，RUN-3，4 は，関連性が高い文書をより多く上位に取得しやすいと判断できる．

次に RUN-3 と RUN-4 を比較する．nDCG@10 の値を比較すると RUN-4 は RUN-3 から 0.092 増加している．図 7 より，RUN-3 と RUN-4 における L2 の文書数に大きな差は存在していないが，RUN-4 の L0 の数は RUN-3 の L0 の数より 76 件増加している．関連性のある文書 L1，L2 の数

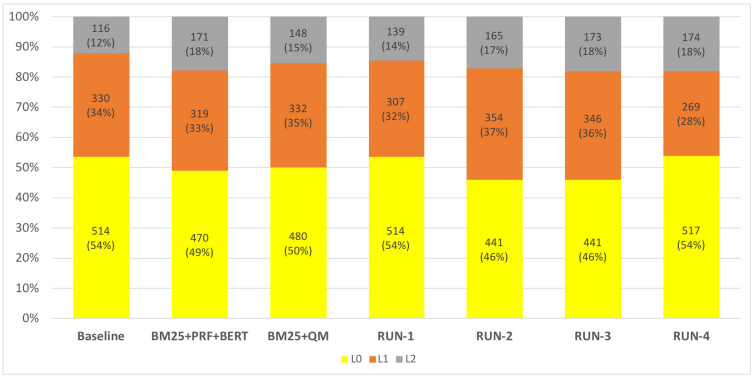


図 7　各手法でランキングされた文書の関連度の内訳
Fig. 7　Breakdown of relevance of ranked documents obtained by each method.

表 12　RUN-3 から RUN-4 に変化させた際の上位 10 件の検索結果の変化
Table 12　Changes in the top 10 search results when changing from RUN-3 to RUN-4.

検索結果の変化の種別	L0	L1	L2	総数
上位 10 件内で順位が上昇した文書数	46	43	13	102
上位 10 件内で順位が下降した文書数	14	13	16	43
順位に変化がない文書数	85	120	62	267
上位 10 件に新規に含まれた文書数	372	93	83	548
上位 10 件から除外された文書数	296	170	82	548

は RUN-3 の方が多いが nDCG@10 では RUN-4 が高い結果から，RUN-4 の方が関連性が高い文書が上位にランキングされている可能性が高い．**表 12** に RUN-3 から RUN-4 へ変化させた際の検索結果の変化を示す．

表 12 の結果から，上位 10 件内で順位が上昇した L1，L2 文書数は 56 件であるが，上位 10 件内で順位が下降した L1，L2 文書数は 29 件となっており，順位が下降した文書数よりも順位が上昇した文書数が多いことが確認できる．この結果から，RUN-2 から RUN-3 にかけて関連性のある文書の順位が下がっていた問題が，クエリ拡張を追加することで改善されている．

最後に，比較手法 BM25+PRF+BERT，BM25+QM と提案手法を比較する．**表 13**，**表 14** にそれぞれ BM25+PRF+BERT と BM25+QM から RUN-4 へ変化させた際の検索結果の変化を示す．表 10〜表 12 の上位 10 件に新規に含まれた L1，L2 文書数と表 13，表 14 の上位 10 件に新規に含まれた L1，L2 文書数を比べると，前者よりも後者の L1，L2 文書数が多いことが確認できる．これは，比較手法で得られた検索結果が提案手法の検索結果と大きく異なることを示唆している．nDCG@10 の値を比較すると，RUN-1 を除くすべての手法で nDCG@10 の値を上回っていることが確認できる．また，図 7 より，BM25+PRF+BERT と BM25+QM における L2 の文書数より RUN-3，4 の L2 の文書数の方が多い．関連性のある文書 L1，L2 の数は BM25+PRF+BERT と BM25+QM の

表 13　BM25+PRF+BERT から RUN-4 に変化させた際の上位 10 件の検索結果の変化
Table 13　Changes in the top 10 search results when changing from BM25+PRF+BERT to RUN-4.

検索結果の変化の種別	L0	L1	L2	総数
上位 10 件内で順位が上昇した文書数	10	12	12	34
上位 10 件内で順位が下降した文書数	9	21	18	48
順位に変化がない文書数	5	6	4	15
上位 10 件に新規に含まれた文書数	493	230	140	863
上位 10 件から除外された文書数	446	280	137	863

表 14　BM25+QM から RUN-4 に変化させた際の上位 10 件の検索結果の変化
Table 14　Changes in the top 10 search results when changing from BM25+QM to RUN-4.

検索結果の変化の種別	L0	L1	L2	総数
上位 10 件内で順位が上昇した文書数	11	13	10	23
上位 10 件内で順位が下降した文書数	10	21	21	42
順位に変化がない文書数	4	7	4	15
上位 10 件に新規に含まれた文書数	492	228	139	859
上位 10 件から除外された文書数	455	291	113	859

方が多いが，nDCG@10 では RUN-4 が高い．これより，RUN-4 の方が関連性が高い文書が上位にランキングされやすいことが考えられる．

提案手法の汎用性について考察する．提案手法では，カテゴリ検索，クエリ拡張にそれぞれ Yahoo! 知恵袋データを利用しており，Yahoo! 知恵袋に依存した側面を持つともいえる．しかし，Yahoo! 知恵袋は，一般ユーザが疑問に思うことを質問できるサイトであり，一般ユーザの情報要求のかなりの部分は網羅していると考えられる．それまで存在しなかった新たな話題の質問が追加される場合においても，運営側が一定期間ごとにカテゴリを更新することで，適切なカテゴリや質問データが利用可能である．一方，Yahoo! 知恵袋には投稿されないような特殊な話題の質問が存在する場合は，そのような質問データやカテゴリ，カテゴリと関連づけられた統計データを本稿で扱った程度の

規模で収集できれば，提案手法を適切に適用することができると考えられる．今後，Yahoo! 知恵袋によらない質問や他のデータセットでの評価をすることで，提案手法の汎用性を検証していきたい．

8. まとめ

本稿では，統計データに対するアドホック検索の問題に対して，被検索文書集合をカテゴリで絞り込むカテゴリ検索手法，統計データ本体から表ヘッダ情報を抽出し，被検索文書の一部として補強する手法，および，クエリ拡張をそれぞれ組み合わせるランキング手法を提案した．評価実験の結果，カテゴリ検索とデータ補強は，それぞれ，nDCG@10 の値を改善し，特に，データ補強のみの手法では nDCG@10 で 0.567 と最も良好な値を示した．

しかし，上位 10 件の検索結果中の関連性のある文書数は，データ補強のみの手法では 165 件なのに対して，カテゴリ検索，データ補強，クエリ検索すべてを組み合わせた手法では 174 件であった．すべて組み合わせた提案手法は nDCG@10 が 0.534 で，データ補強のみの手法よりわずかに値は低いが，上位 10 件内での関連性が高い文書数では，データ補強のみの手法よりも多く，関連性が高い文書をより多く上位に取得しやすいことを確認した．

また，すべて組み合わせた提案手法は，既存研究の nDCG@10 の値を上回り，関連性が高い文書数でも上回っていることを確認した．

今後は，統計文書の内容を補強する手法やランキング手法などについて改善を図っていく予定である．また，Yahoo! 知恵袋によらない質問や他のデータセットでの評価をすることで，提案手法の汎用性を検証していく予定である．

謝辞　本研究の一部は科研費 18K11557 の助成を受けたものである．また，本研究では，国立情報学研究所の IDR データセット提供サービスによりヤフー株式会社から提供を受けた「Yahoo! 知恵袋データ（第 3 版）」を利用した．ここに記して感謝の意を表します．

参考文献

[1] Kato, M.P., Ohshima, H., Liu, Y.-H. and Chen, H.-L.: Overview of the NTCIR-15 data search task, *Proc. NTCIR-15 Conference* (2020).

[2] Raghavan, V.V. and Wong, S.K.M.: A critical analysis of vector space model for information retrieval, *Journal of the American Society for Information Science*, Vol.37, No.5, pp.279–287 (1986).

[3] Fuhr, N.: Probabilistic Models in Information Retrieval, *The Computer Journal*, Vol.35, No.3, pp.243–255 (1992).

[4] Turtle, H. and Croft, W.B.: Evaluation of an inference network-based retrieval model, *ACM Trans. Information Systems* (*TOIS*), Vol.9, No.3, pp.187–222 (1991).

[5] Manning, C.D., Raghavan, P. and Schutze, H.: *Introduction to information retrieval*, Cambridge University Press, Cambridge (2008).

[6] Shi, P., Rao, J. and Lin, J.: Simple attention-based representation learning for ranking short social media posts, *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1* (*Long and Short Papers*), pp.2212–2217, Association for Computational Linguistics (2019).

[7] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding, *Proc. 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, Association for Computational Linguistics (2019).

[8] Yilmaz, Z.A., Wang, S., Yang, W., Zhang, H. and Lin, J.: Applying BERT to document retrieval with birch, *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*): *System Demonstrations*, pp.19–24, Association for Computational Linguistics (2019).

[9] Zhang, S. and Balog, K.: Ad hoc table retrieval using semantic similarity, *Proc. 2018 World Wide Web Conference*, *WWW '18*, pp.1553–1562, Republic and Canton of Geneva, CHE 2018, International World Wide Web Conferences Steering Committee (2018).

[10] Shraga, R., Roitman, H., Feigenblat, G. and Canim, M.: Ad hoc table retrieval using intrinsic and extrinsic similarities, *Proc. Web Conference 2020*, *WWW '20*, pp.2479–2485, Association for Computing Machinery (2020).

[11] Chen, Z., Trabelsi, M., Heflin, J., Xu, Y. and Davison, B.D.: Table search using a deep contextualized language model, *Proc. 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, *SIGIR '20*, pp.589–598, Association for Computing Machinery (2020).

[12] 中野　優，加藤　誠：誤引用検証ための被引用統計データの検索，第 13 回データ工学と情報マネジメントに関するフォーラム（DEIM2021），F25-4 (2021).

[13] Watarai, T. and Tsuchiya, M.: Developing dataset of Japanese slot filling quizzes designed for evaluation of machine reading comprehension, *Proc. 12th Language Resources and Evaluation Conference*, pp.6895–6901, European Language Resources Association (2020).

[14] Shimbun, M.: Cd-mainichi shimbun 1995 data collection, nichigai associates (1996).

[15] Nguyen, P., Shinoda, K., Sakamoto, T., Petrescu, D.A., Tran, H.N., Takasu, A., Aizawa, A. and Takeda, H.: Nii table linker at the ntcir-15 data search task: Re-ranking with pre-trained contextualized embeddings, data content, entity-centric, and cluster-based approaches, *Proc. NTCIR-15 Conference* (2020).

[16] Mibayashi, R., HuuLong, P., Matsumoto, N., Yamamoto, T. and Ohshima, H.: Uhai at the ntcir-15 data search task, *Proc. NTCIR-15 Conference* (2020).

岡本 卓　（学生会員）

2019 年京都産業大学コンピュータ理工学部インテリジェントシステム学科卒業．現在，同大学大学院先端情報学研究科修士課程 1 年在学．主に，情報検索，自然言語処理，質問応答に関する研究に従事．日本データベース学会会員．

宮森 恒　（正会員）

1992 年早稲田大学理工学部電子通信学科卒業，1994 年同大学大学院理工学研究科修士課程修了，1997 年同大学大学院理工学研究科後期博士課程修了．1996～1997 年同大学理工学部助手．1997 年郵政省通信総合研究所入所．独立行政法人情報通信研究機構主任研究員サブグループリーダ兼務を経て，2008 年京都産業大学コンピュータ理工学部准教授．現在，同大学大学院先端情報学研究科教授．工学博士．主に，マルチメディアデータ工学，パターン認識，情報検索に関する研究に従事．2006 年日本データベース学会平成 17 年度論文賞．ACM，電子情報通信学会，日本データベース学会，人工知能学会，言語処理学会，映像情報メディア学会各会員．

（担当編集委員　関 和宏）