DEIM Forum 2021 F31-5

カテゴリ分類とメタデータ補強に基づく 統計データに対するアドホック検索

岡本　卓† 宮森　恒†

† 京都産業大学大学院 先端情報学研究科　〒 603–8555 京都府京都市北区上賀茂本山
E-mail: †{i2086042,miya}@cc.kyoto-su.ac.jp

あらまし　本稿では，カテゴリ分類とメタデータ補強に基づく，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，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対象文書とは異なる特徴をもつ．本稿では，被検索対象文書群とクエリをカテゴリ分類することで，候補となる被検索対象文書を絞り込み，表本体のヘッダ情報で補強された統計データのメタデータとクエリ拡張を用いてランキングする手法を提案する．実験では，いくつかのベースライン手法と比較し，提案手法の有用性を検証する．

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

1 はじめに

本稿では，従来の自然言語で記述されたテキスト文書に対するアドホック検索とは異なり，各国政府などから提供されている公共的に利用可能なオープンデータ等のデータに対するアドホック検索に取り組む．情報検索技術の発展と普及により，画像や動画，音楽などのデータを検索し利用することはすでに日常的となっている．一方，各国政府や公共機関などが提供しているオープンデータについては，ユーザの情報要求を適切に満たすアドホック検索は必ずしも実現されているとはいえない．オープンデータは，世界規模の課題を，世界中の人々が協力して取り組むのに不可欠なリソースの一つであり，データ検索は近年多くの研究者に注目されている．

本稿では，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，日本政府による統計データ (e-Stat) および米国政府による統計データ (Data.gov) それぞれを対象としたアドホック検索の課題に取り組むこととした．

扱う対象文書データセットは，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，数値で構成されているという特徴がある．

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捉えるために，被検索文書集合をカテゴリに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，被検索文書とする手法，および，クエリ拡張を組み合わせた手法を提案する．さらに，クエリから単語を生成してクエリに追加する手法を取ることで，ヘッダから抽出した情報を活用して通常の検索では上位に来にくい統計データを検索できることに期待する．

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，文書にスコアをつける手法がいくつも提案されてきた．手法には，単語や文書，クエリをベクトルに変換して空間ベクトルとして計算するベクトル空間モデル [1] やクエリと文書の関連する確率を計算する確率モデル [2]，ニューラルネットを使用した推論ネットワークモデル [3] などがある．特に確率モデルの一種である BM25 [4] は現在でも高い精度でクエリに関連する文書を返すことができる．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 [5] 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の BERT [6] を使用した検索手法 [7] も提案されている．

オープンデータを対象とした検索に関するカンファレンス [8] も開催された．カンファレンスでは，クエリの理解やデータ構造の理解，検索モデルの発展を目的として，統計データに対する検索タスクが与えられ，これまで，入力されたクエリを統計データを検索するのに最適なクエリに修正する手法 [9] や，入力クエリと統計データの関連度を BERT を使用して学習して再ランキングする手法 [10]，被検索統計データをクラスタリングしてクエリに近いクラスタを対象に検索をする手法 [11] などが提案された．提案された手法の多くで事前学習モデル BERT が利用されていたが，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の精度を大きく超える結果には至っていない．

3 データセット

本稿で検索する文書は，統計データセット中のメタデータとそれに対応する 1 つ以上の統計データで構成される．統計デー

タセットは日本政府による統計データ (e-Stat) および米国政府による統計データ (Data.gov) それぞれを対象に収集したものである．統計データのファイル形式の内訳を表 1 に示す．

表 1　**統計データのファイル形式の分布**

(a) 日本語 (e-Stat)

ファイル形式	頻度
xls	686436
csv	568042
pdf	49124
xlsx	34794
xlsm	6

(b) 英語 (Data.gov)

ファイル形式	頻度
pdf	47260
x_gzip	17099
html	9296
xml	4443
csv	2919
plain	2761
none	2542
json	1938
rdf+xml	1484
octet-stream	1430
ms-excel	753
sheet	568
zip	98
ms-word	88
pgp-signature	55
document	54
x-octet-stream	37
x-zip-compressed	15
rss+xml	11
x-zip	11
x-sh	7
excel	1
pcap	1
javascript	1
jpeg	1

メタデータは，e-Stat，Data.gov それぞれの統計データの導入ページから抽出されたデータであり，統計データはメタデータと紐づけられた表形式の統計データ本体である．メタデータは JSON 形式であり，id や url, タイトルのほか，統計データを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値で構成されている．表 2 にメタデータの title，description の値の単語数の平均と標準偏差，および，メタデータと統計データの総数を示す．

表 2　統計データセットにおける文書 (メタデータ、統計データ) の諸元

言語	メタデータの title および description の単語長		メタデータの総数	統計データの総数
	平均	標準偏差		
英語	101.93	81.19	46,615	92,930
日本語	11.83	3.02	1,338,402	1,338,402

それに対して従来のアドホック検索で用いられる新聞記事の文書長は，英語で 400 単語 [12]，日本語で 330 単語 [13] 程度である．よって表 2 と比較するとメタデータの文書長は従来のアドホック検索の文書長よりも短いという特徴がある．

そのため，従来のアドホック検索手法をそのまま用いただけでは，クエリを適切に満たす検索結果を得ることは難しい．そこで，クエリからカテゴリを判別することで，被検索文書を絞り込む手法を提案する．また，統計データ本体を参照しメタデータ情報を補強する手法，および，クエリ拡張を組み合わせた手法を提案する．

4　カテゴリ検索

統計データ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捉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，予め構築した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 1 に示す．

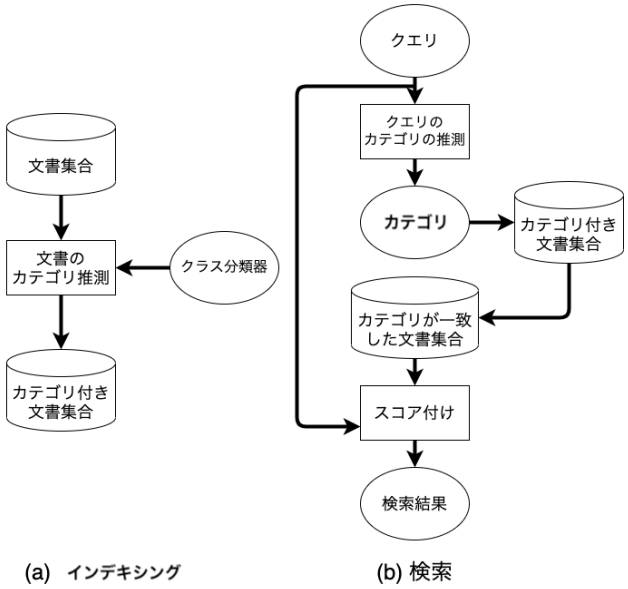

図 1　カテゴリ検索の概要

以下，処理手順を詳しく説明する．まず，カテゴリ集合を次のように定義する．

$$C = \{c_p\} \tag{1}$$

インデキシング時には，各被検索文書 d_j には，テキスト分類器 $text\_classifier$ により，あるカテゴリ c_p がラベル l_j として付与される．

$$(d_j, l_j), \quad l_j = c_p = text\_classifier(d_j) \in C \tag{2}$$

ラベルが付与された被検索文書集合 D' が索引付けされ登録される．

$$D' = \{(d_j, l_j)\} \tag{3}$$

検索時には，クエリ q_i に対して，テキスト分類器

$text\_classifier$ を用いてカテゴリ c_p を推定する．

$$c_p = text\_classifier(q_i) \in C \tag{4}$$

カテゴリ c_p に属する被検索文書集合 D_{c_p} に対してのみ関数 $ranking$ によりランキングを実行し，検索結果 R_i を返す．

$$D_{c_p} = \{d_j | (d_j, l_j) \in D',\ l_j = c_p \in C\} \tag{5}$$

$$R_i = ranking(q_i, D_{c_p}) \tag{6}$$

ただし，検索結果 R_i が閾値 θ に対して $|R_i| < \theta$ となる場合は，元の被検索文書集合 D に対する検索結果を追加する．

$$R_i = ranking(q_i, D_{c_p}) \cup ranking(q_i, D) \tag{7}$$

カテゴリ集合 C として，日本語では，コミュニティ質問応答 Web サービス Yahoo! 知恵袋で用いられている 10 種類を，英語では，Yahoo! Answers で用いられている 23 種類を用いることとした．

表 3　カテゴリ検索で使用したカテゴリと収集した QA ペア数

(a) Yahoo! 知恵袋

カテゴリ	# of QA
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

(b)Yahoo! Answers

カテゴリ	# of QA
art	260
business	260
cars	260
computer	20
dining	89
education	58
entertainment	220
family	260
food	20
games	260
health	260
home	54
local	180
news	240
pets	240
politics	160
pregnancy	20
science	260
social	140
society	60
sports	260
style	40
travel	20

また，被検索文書やクエリをカテゴリに分類するために，テキスト分類器 $text\_classifier$ を構築する．Yahoo! 知恵袋，および，Yahoo! Answers の各カテゴリの質問と回答ペアのうち，回答に URL を含んでいるものを収集した．表 3 に，カテゴリごとの収集した質問・回答ペアの件数を示す．各カテゴリの質問・回答ペアの件数は，Yahoo! 知恵袋で平均 149.4，標準偏差 6.3 であり，Yahoo! Answers で平均 158.3，標準偏差 99.3 である．カテゴリによっては質問・回答ペアの件数が少ないものも存在したが，1 カテゴリあたり少なくとも 20 件以上となるように収集した．

また，分類器を構築するために，収集した質問と回答ペアについて，全品詞, 名詞, 名詞＋動詞の 3 通りに対して，skip-gram, GloVe, fastText, 各単語の頻度 (tf) の 4 種類の方法でベクトルに変換し，SVM，ロジスティック回帰 (LR)，ランダムフォレスト，多層ニューラルネットワーク, 多層パーセプトロン (MLP) の 5 通りの方法でそれぞれ学習させた．

以上の要素の全ての組み合わせで 10 交差検証したところ，日本語では，名詞＋動詞，fastText，SVM と LR の組み合わせの際，最も高い分類精度 0.69 を，英語では，全品詞，TF，MLP の組み合わせの際，最も高い分類精度 0.66 となった．

5　ヘッダ抽出

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．

具体的には，統計データ内の各セルについて文字が含まれるか否かを調べ，文字を含むセル数の変化によってヘッダを抽出する．抽出手順をアルゴリズム 1 に示す．まず，直前の空でないセル数 $prev$ を 0 で，列ヘッダ hdr_col を空のリストで初期化し，統計データの列数を max_col に格納する．1 列目から max_col 列まで以下を繰り返す．現在の列の空でないセル数 $curr$ が $prev$ よりも大きければ，その列の空でないセルを列ヘッダ hdr_col に追加する．$prev$ を $curr$ で更新し，次の列の処理に移行する．繰り返しが終了したら，列ヘッダ hdr_col を h_j^{col} に返す．行ヘッダについては，アルゴリズム 1 における列を行に置き換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 に連結することで，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．

Algorithm 1 Extracting column headers from statistical data

Input: statistical data sd
Output: column headers hdr_col

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

一方，英語については，統計データのほとんどが pdf で構成されているため，日本語と同様にセルにアクセスすることが困難である．よって，英語では，PDFminer と呼ばれるモジュールを用いて pdf 内の文字情報を全て抽出することとした．全て抽出することで，数値データといった検索時に直接有用でない

情報も取得することになるが，pdf の統計データでは，統計表の解説文が含まれることも多く，検索時に有用となる可能性もある．

6 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 サイトにおける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，これに基づき単語をベクトル化する．日本語データについては，NII[1]から質問タイトルと質問本文を抽出した．英語データについては，4 節で提示したデータに加え，Yahoo! Answers から別途 QA データを収集し，質問タイトルと質問本文を抽出した．両データについて，4 節で提示したデータとの重複を排除し，データセットとした．内訳を表 4 に示す．作成したデータセットから，skip-gram モデルで単語ベクトルに変換する．モデルへの入力には，作成したデータセットの内容語のみを使用する．これは，クエリとして使われる単語は基本的に内容語で構成されると考えられるためである．クエリ拡張をする際は，クエリの各単語について，変換された単語ベクトルのコサイン類似度が最大となる 1 単語を取得し，拡張語としてクエリに追加する．

表 4　skip-gram の学習に使用したデータセット

収集元	データ数
Yahoo! 知恵袋	5,339,061
Yahoo! Answers	4,821

検索時には，クエリ q_i に対して，拡張されたクエリ q_i' を用いてランキングを実行し，検索結果 Ri を返す．ここで，qe_i は，クエリ q_i から得られた拡張語群を表す．

$$q_i' = \{q_i \cup qe_i\} \tag{8}$$

$$R_i = ranking(q_i', D_{cp}) \tag{9}$$

7 評価実験

ここでは，提案手法の様々な組み合わせで，統計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．

7.1 実験方法

検索に使用するクエリは，Yahoo! 知恵袋において，e-Stat へのリンクをもつ QA ペアの質問文から作成する [8]．英語では日本語で作成したクエリを英語に翻訳したものを使用する [8]．作成された各クエリから得られた検索結果のうち上位 10 文書を取得する．上位 10 文書の各文書と，クエリの作成元となった質問文のペアについて，L0,L1,L2 で関連性を評価した（L0：関連性がない，L1：部分的に関連性がある，L2：関連性が高い）．なお，評価には，クラウドソーシングを利用し，日本語では Lancers，英語では Amazon Mechanical Turk を利用した．ワーカには，慎重に質問文を読んだ上で，必ずペアとなっている文書 (統計データ) の内容を参照してから評価を行うように指示した．

なお，Baseline には一般的に使用されている BM25 を選択した．

7.2 評価対象とする手法

実験では，カテゴリ検索のためのテキスト分類器の構成方法や，表のヘッダ情報の取得方法，クエリ拡張の有無で組み合わせた手法を比較した．表 5 に，実験で評価対象とする手法の一覧を示す．なお，クラウドソーシングでの評価は，予備実験で検索結果が良好と判断された組み合わせの手法のみを評価した．また，ベースラインとしては，BM25 によるランキングのみの手法を採用した．

まず，日本語の場合について述べる．

RUN-J-1 は，メタデータの title および description 変数の値で構成される mj からなる被検索文書集合 Dm 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-J-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，行と列の表ヘッダ h_j^{row} と h_j^{col} が追加された被検索文書集合 $D^{m+h^r+h^c}$ に対して，名詞+動詞, fastText, SVM で構築した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．

$$D^{m+h^r+h^c} = \{d_j^{m+h^r+h^c}\} = \{m_j \cup h_j^{row} \cup h_j^{col}\} \tag{10}$$

$$D_{c_p}^{m+h^r+h^c} = \{d_j | (d_j, l_j) \in D^{m+h^r+h^c}, \; l_j = text\_classifier_{N+V,fasttext,SVM}(q_i) \in C\} \tag{11}$$

$$R_i = ranking_{BM25}(q_i, D_{c_p}^{m+h^r+h^c}) \tag{12}$$

RUN-J-3 は，RUN-J-2 の被検索集合 $D^{m+h^r+h^c}$ を以下の被検索集合 D^{m+h^r} に置き換えた手法である．

$$D^{m+h^r} = \{d_j^{m+h^r}\} = \{m_j \cup h_j^{row}\} \tag{13}$$

RUN-J-4 は，RUN-J-2 の被検索集合 $D^{m+h^r+h^c}$ を被検索集合 D^m に置き換えた手法である．

RUN-J-5,6,7 は，RUN-J-2,3,4 で用いたテキスト分類器の学習方法を，SVM ではなくロジスティック回帰でそれぞれ置き換えた手法である．

RUN-J-8 は，予備実験で，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判断された RUN-J-2 の内容に，クエリ拡張を追加した手法である．

1：国立情報学研究所 Yahoo! 知恵袋データ（第 3 版）：
https://www.nii.ac.jp/dsc/idr/yahoo/chiebkr3/Y_chiebukuro.html

表 5　実験で評価した手法の一覧

name	category search	table header	query extension	ranking	text classifier for category search			table header extraction
					POS	vector	training	
Baseline-J				BM25				
RUN-J-1	✓			BM25	All	TF	MLP	
RUN-J-2	✓	✓		BM25	N+V	fastText	SVM	ROW+COL
RUN-J-3	✓	✓		BM25	N+V	fastText	SVM	ROW
RUN-J-4	✓			BM25	N+V	fastText	SVM	
RUN-J-5	✓	✓		BM25	N+V	fastText	LogisticRegression	ROW+COL
RUN-J-6	✓	✓		BM25	N+V	fastText	LogisticRegression	ROW
RUN-J-7	✓			BM25	N+V	fastText	LogisticRegression	
RUN-J-8	✓	✓	✓	BM25	N+V	fastText	SVM	ROW+COL
Baseline-E				BM25				
RUN-E-1	✓			BM25	All	TF	MLP	
RUN-E-2	✓	✓		BM25	All	fastText	SVM	All
RUN-E-3	✓			BM25	All	fastText	SVM	
RUN-E-4	✓	✓		BM25	All	fastText	LogisticRegression	All
RUN-E-5	✓			BM25	All	fastText	LogisticRegression	
RUN-E-6	✓		✓	BM25	All	TF	MLP	

次に，英語の場合について述べる．

RUN-E-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-E-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，統計データ全体 t_j が追加された被検索文書集合 D^{m+t} に対して，名詞+動詞, fastText, SVM で構築した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．

$$D^{m+t} = \{d_j^{m+t}\} = \{m_j \cup t_j\} \tag{14}$$

$$D_{c_p}^{m+t} = \{d_j | (d_j, l_j) \in D^{m+t}, \\ l_j = text\_classifier_{all,fasttext,SVM}(q_i) \in C\}$$

$$R_i = ranking_{BM25}(q_i, D_{c_p}^{m+t}) \tag{15}$$

RUN-E-3 は，RUN-E-2 の被検索集合 D^{m+t} を被検索集合 D^m に置き換えた手法である．

RUN-E-4,5 は，RUN-E-2,3 で用いたテキスト分類器の学習方法を，SVM ではなくロジスティック回帰でそれぞれ置き換えた手法である．

RUN-E-6 は，予備実験で，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判断された RUN-E-1 の内容に，クエリ拡張を追加した手法である．

7.3 実験結果

各手法の評価結果を表 6 に示す．

まず，日本語での結果について分析する．カテゴリ検索と BM25 を組み合わせた手法である RUN-J-1 では他の手法に比べて比較的高い値が得られた．しかし，内訳を確認したところ，多くのクエリや文書が誤ったカテゴリに分類されていることがわかった．そのような中で比較的高い nDCG@10 の値が得られたのは，誤ったカテゴリに関連文書が多く存在しており，結果的に被検索文書の絞り込みが有効に機能したことが考えられる．一方，頻度ベクトルを使ったことは，日本語の多様な表現に対応するには必ずしも適切ではなく，誤判別につながった要因の一つと考えられる．

表 6　評価結果

RUN	nDCG@10
Baseline-J	0.386
RUN-J-1	0.413
RUN-J-2	0.426
RUN-J-3	0.276
RUN-J-4	0.353
RUN-J-5	0.426
RUN-J-6	0.276
RUN-J-7	0.342
RUN-J-8	**0.533**
Baseline-E	0.232
RUN-E-1	**0.240**
RUN-E-2	0.042
RUN-E-3	0.181
RUN-E-4	0.043
RUN-E-5	0.216
RUN-E-6	0.093

一方，RUN-J-4,7 のカテゴリ分類器の分類精度を 10 交差検証で求めると 0.68 であり，RUN-J-1 の場合の 0.23 と比較すると大きく改善していることが確認された．このような状況が生じた要因として，RUN-J-4,7 では，より適切なカテゴリに分類されるようになったものの，絞り込まれた文書群中に，情報要求を満たす適切な関連文書があまり存在していなかったことが挙げられる．実際,「23 歳睡眠時間」というクエリは，RUN-J-1 では「生活」，RUN-J-4,7 では「健康」というカテゴリに分類されるが,「健康」カテゴリで絞り込まれた文書はほとんどが医

療関連の内容となっており，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事例を確認できた．

RUN-J-4,7 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-J-2,5 による nDCG@10 の値は，RUN-J-4,7 による値と比較して，それぞれ 0.073, 0.084 増加した．同様に，RUN-J-4,7 での被検索文書集合を，行ヘッダのみで補強した文書集合とした KSU-J-3,6 による nDCG@10 の値は，RUN-J-4,7 による値と比較して，それぞれ 0.077, 0.066 減少した．統計データの行ヘッダと列ヘッダを用いてメタデータを補強することはより適切なランキングにつなげるために有用であると考えられる．

一方，行ヘッダのみによる補強は適切なランキングにつながらなかった．実際，行ヘッダの内容を確認すると，図 2 に示すように，検索に直接役立つとは思えない文字列が含まれている例を確認することができた．この例の場合，1 列目と 2 列目の空でないセルの行数が同じであるため,「HS」を含む列が行ヘッダとして抽出されるが,「全国」などの地名を含む文字列は現在は行ヘッダとして抽出されていない．ヘッダをより的確に抽出するようにアルゴリズムを改善する必要がある．

HS	00　全　国	379297	300059	7581
HS	01　北海道	6157	4919	131
HS	01100札幌市	-	-	-
HS	01202函館市	-	-	-
HS	01203小樽市	-	-	-
HS	01204旭川市	-	-	-
HS	01205室蘭市	-	-	-
HS	01206釧路市	-	-	-
HS	01207帯広市	-	-	-
HS	01208北見市	-	-	-
HS	01209夕張市	-	-	-
HS	01210岩見沢i	-	-	-
HS	01211網走市	-	-	-
HS	01212留萌市	-	-	-
HS	01213苫小牧i	130	112	-
HS	01214稚内市	-	-	-
HS	01215美唄市	157	123	1

図 2　適切な行ヘッダが取得できない例

RUN-J-2 にクエリ拡張を追加した RUN-J-8 による nDCG@10 の値は，RUN-J-2 と比べて 0.107 増加した．クエリ拡張により適切な統計データを関連づけられるようになったことがわかる．これについては 8 節で考察する．

次に，英語での結果について分析する．

カテゴリ検索と BM25 を組み合わせた手法である RUN-E-1 では，他の手法に比べて比較的高い値が得られた．しかし，RUN-J-1 と同様に，多くのクエリや文書が誤ったカテゴリに分類されるものの，誤ったカテゴリに関連文書が多く存在しており，結果的に被検索文書の絞り込みが有効に機能したことが考えられる．

RUN-E-1 でのカテゴリ分類器の構築方法を変えた RUN-E-3,5 による nDCG@10 の値は，RUN-E-1 による値と比較して，それぞれ 0.059, 0.024 減少した．一方，RUN-E-3,5 のカテゴリ分類器の分類精度を 10 交差検証で求めると 0.42 であり，RUN-J-1 の場合の 0.53 と比較すると減少していることが確認された．カテゴリの分類精度が減少した要因として，学習に使用したカテゴリごとのデータ数の偏りが影響した可能性が考えられる．表 3 に示す通り，Yahoo!知恵袋から収集したカテゴリごとのデータ数の最大は 163，最小は 140 であったが，Yahoo!Answers から収集したカテゴリごとのデータ数の最大は 260，最小は 20 である．fastText による分類器は，ターム頻度ベクトルよる分類器よりも，より忠実にこれらデータ数の偏りの影響を学習し，結果的に分類精度が減少したことが考えられる．カテゴリごとのデータ数の偏りを是正することでスコアを改善できる可能性があると考えられる．

RUN-E-3,5 の被検索文書集合を，統計データ全体で補強した文書集合とした RUN-E-2,4 による nDCG@10 の値は，RUN-E-3,5 による値と比較して，それぞれ 0.157,0.173 減少した．RUN-E-3,5 では，統計データ全体で補強しているため，数値データなどの，検索に直接有用とはなりにくい情報が多く含まれていたため，ランキングに悪影響を及ぼしたと考えられる．実際，文書の内容を確認すると，数値データや統計表を扱う上での注意事項といった，クエリに直接関係しそうにない情報が多数確認された．ヘッダのみを的確に抽出する手法を適用することでスコアを改善できる可能性が考えられる．

RUN-E-1 にクエリ拡張を追加した RUN-E-6 による nDCG@10 の値は，RUN-1-E と比べて 0.147 減少した．英語の場合は，クエリ拡張したことにより適切な統計データへの関連づけが阻害されていることがわかる．これについては 8 節で考察する．

8　考　　察

本節では，nDCG@10 の値の改善と悪化が顕著であったクエリ拡張について考察する．

まず，クエリ拡張の有無によって，検索結果に含まれる文書の順位がどのように変化したかを確認した．96 クエリについて，クエリ拡張を追加した手法 (RUN-J-8, RUN-E-6) と追加しなかった手法 (RUN-J-2, RUN-E-1) のそれぞれについて，上位 10 件の検索結果文書にみられた変化を表 7 に示す．

日本語の場合については，上位 10 件内での順位が上昇した文書のうち，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 54 件，48 件，上位 10 件内での順位が下降した文書のうち，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 23 件，20 件であり，相対的に L2,L1 評価の文書数が上昇したことが示唆される．また，クエリ拡張なしの場合，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 226 件，186 件 (L2,L1 評価の文書数の割合は 54.9%) であるが，クエリ拡張ありの場合，L2,L1 評価の文書数と L0 評価の文書数は 393 件，567 件 (L2,L1 評価の文書数の割合は 40.9%) となり，L2,L1 評価の文書数の割合自体は低下している．クエリ拡張により，L2,L1 評価の文書の順位が上昇したことにより nDCG@10 の値が改善したことが示唆される．

英語の場合については，上位 10 件内での順位が上昇した文書のうち，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 2

件，4 件，上位 10 件内での順位が下降した文書のうち，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 0 件，3 件であり，相対的に L2,L1 評価の文書数がわずかに上昇したことが示唆される．また，クエリ拡張なしの場合，L2,L1 評価の文書数と L0 評価の文書数はそれぞれ 3 件，9 件 (L2,L1 評価の文書数の割合は 25.0%) であるが，クエリ拡張ありの場合，L2,L1 評価の文書数と L0 評価の文書数は 45 件，915 件 (L2,L1 評価の文書数の割合は 4.7%) となり，L2,L1 評価の文書数の割合自体は大きく低下している．クエリ拡張により，L0 評価の文書の割合が非常に大きくなったことで nDCG@10 の値が悪化したことが示唆される．

表 7　クエリ拡張を追加したことによる上位 10 件の検索結果の変化

	検索結果の変化の種別	L0	L1	L2	総数
日本語	上位 10 件内で順位が上昇した文書数	48	31	23	102
	順位が下降した文書数	20	12	11	43
	上位 10 件内で順位が下降した文書数	118	107	42	267
	上位 10 件に新規に含まれた文書数	381	82	85	548
	上位 10 件から除外された文書数	299	167	82	548
英語	上位 10 件内で順位が上昇した文書数	4	2	0	6
	上位 10 件内で順位が下降した文書数	3	0	0	3
	順位に変化がない文書数	2	1	0	3
	上位 10 件に新規に含まれた文書数	906	39	3	948
	上位 10 件から除外された文書数	808	130	10	948

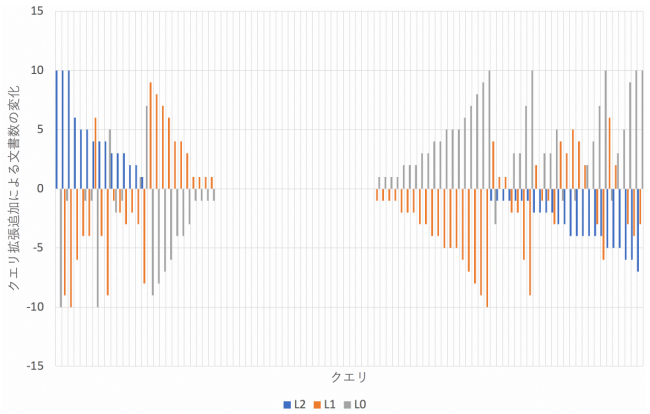

図 3　クエリ拡張追加による，クエリごとの
各評価の文書数の変化 (RUN-J-2,8)

次に，クエリ拡張を追加することで，L0,L1,L2 の文書数がどのように変化したかを，96 クエリのそれぞれについて算出した結果を図 3,4 に示す．

まず，改善されたクエリを確認したところ，クエリ拡張により，より適切な関連文書の取得につながっている事例を確認した．元のクエリが ” 学習塾 割合 小学生から高校生 ” であるとき，クエリ拡張により ” 小学生から高校生 ” から ” 中学生 ” という単語が追加された．これにより，” 小学生，中学生，高校生学習塾の通学割合 ” というタイトルの統計データの文書がより上位にランキングされていることを確認した．

次に，L2 評価の文書数が減少し，検索結果が悪化したクエリを確認した．クエリが ” 小田原市 東京 通勤 人数 ” のとき，クエリ拡張により ” 東京 ” から ” 大阪 ” という単語が追加された．これにより，地域による絞り込みがより厳しく適用され，結果的に該当する統計データの数が少なくなることで，検索結果が悪化したと考えられる．

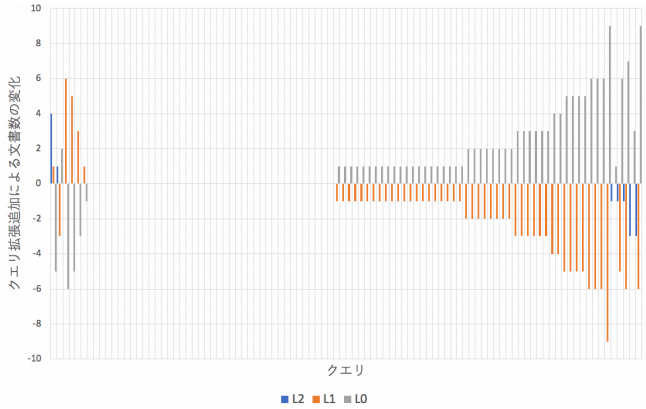

図 4　クエリ拡張追加による，クエリごとの
各評価の文書数の変化 (RUN-E-1,6)

英語については，上位 10 件内で順位が変化した文書が，日本語の場合と比べて非常に少ない．変化が見られたのは，L1 評価の文書数の増加のみで，L2 評価の文書数には変化がみられなかった．さらに，上位 10 件に新規に含まれた文書数のうち，L2 評価の文書数はわずかであり，ほとんどが L0 評価の文書であったこともあり，nDCG@10 の値が減少したと考えられる．実際，クエリ拡張で得られた拡張語を確認すると，全体的に元のクエリとはずれた単語が追加されている．例えば，元のクエリ ” married ” から ” LGBT ” などが取得されており，現在提供されている統計データにはまだほとんど反映されていないと思われる単語の場合，該当する統計データの文書が存在せず，結果的に検索結果の悪化を招いたことが示唆される．

特に，英語でのクエリ拡張が失敗した要因として，学習に使用した Yahoo! Answers の QA データセットは，カテゴリごとのデータ数のばらつきが大きいことが挙げられる．データセット全体のデータ数は一定の規模で収集できているが，カテゴリごとのデータ数はかなり大きくばらついており，学習の際にこのばらつきの影響を大きく受けている可能性が高い．今後は，偏りのない QA データセットを作成していく必要がある．

9　結　　論

本稿では，被検索文書集合をカテゴリで絞り込むカテゴリ検索，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して補強する手法，および，クエリ拡張を組み合わせる手法を提案した．評価実験の結果，カテゴリ検索，ヘッダ抽出，クエリ拡張を組み合わせた提案手法が，日本語において，nDCG@10 で 0.533 と最も良好な値を示した．

しかし，一般的なアドホック検索では 0.9 を超えており，提案手法の性能はまだ十分であるとは言いがたい．今後は，英語におけるヘッダ抽出アルゴリズムを改良し，精度を改善することなど，統計データを補強する手法の改善が主な課題である．

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．
ここに記して感謝の意を表します．

文　　献

[1] Vijay V. Raghavan and S. K. M. Wong. 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.

[2] Norbert Fuhr. Probabilistic Models in Information Retrieval. *The Computer Journal*, Vol. 35, No. 3, pp. 243–255, 06 1992.

[3] Howard Turtle and W. Bruce Croft. Evaluation of an inference network-based retrieval model. 1991.

[4] Christopher D.Manning,Prabhakar Raghavan,Hinrich Schutze. *Introduction to information retrieval*. Cambridge University Press, Cambridge, 2008.

[5] Peng Shi, Jinfeng Rao, and Jimmy Lin. Simple attention-based representation learning for ranking short social media posts. *CoRR*, Vol. abs/1811.01013, , 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, Vol. abs/1810.04805, , 2018.

[7] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying bert to document retrieval with birch, 2019.

[8] Makoto P. Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. Overview of the NTCIR-15 data search task. In *Proceedings of the NTCIR-15 Conference*, 2020.

[9] Ryota Mibayashi, Pham HuuLong, Naoaki Matsumoto, Takehiro Yamamoto, and Hiroaki Ohshima. Uhai at the ntcir-15 data search task. In *Proceedings of the NTCIR-15 Conference*, 2020.

[10] Lya Hulliyyatus Suadaa, Lutfi Rahmatuti Maghfiroh, Isfan Nur Fauzi, and Siti Mariyah. Stis at the ntcir-15 data search task: Document retrieval re-ranking. In *Proceedings of the NTCIR-15 Conference*, 2020.

[11] Phuc Nguyen, Kazutoshi Shinoda, Taku Sakamoto, Diana Andreea Petrescu, Hung Nghiep Tran, Atsuhiro Takasu, Akiko Aizawa, and Hideaki Takeda. 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. In *Proceedings of the NTCIR-15 Conference*, 2020.

[12] Inches G, Carman M, and Crestani F. *Advances in Information Retrieval*. Statistics of online user-generated short documents, 2010.

[13] Mainichi Shimbun. Cd-mainichi shimbun 1995 data collection, nichigai associates, 1996.