E-028

タイトルのうろ覚えを許容する書籍検索

Book Search Allowing for Incomplete Title Recall

池田 桃果‡ 伊東 嗣功‡ 石井 雅樹‡ 堂坂 浩二‡
Momoka Ikeda Hidekatsu Ito Masaki Ishii Kohji Dohsaka

1. はじめに

日頃，どこかで見聞きしたものを調べたいと思ったとき，その名称がうろ覚えになっている場合も多い．そのようなとき，多くの人が利用するのは検索サイトであるが，本を探しているのに楽曲が出てくるなど，同じ言葉で別のジャンルの全く関係のない検索結果になることや，重要視してほしい単語が重要視されないため，求めている検索結果にならないことがある．そのため，検索語の工夫のため不要な時間を要したり，調べたかったものを見つけ出すのは困難となることが考えられる．また，書籍や楽曲等のジャンルごとの検索システムではあいまい検索ができるシステムが少なく，スペルミスやうろ覚えなどが原因でうまく求めていた結果が出ないことも多い．そのような場面の一部に，書籍をテレビや新聞・インターネットで見かけ，読みたいと本屋や図書館に足を運んだとき，タイトルや著者がうろ覚えになっていることがあげられる．しかし，図書館や書店で提供されている蔵書検索サービスは完全一致の検索しかできないものも多く，うろ覚えの書籍を検索することは困難である．

このとき，有用なサービスの 1 つとして，図書館で提供されているレファレンスサービスが挙げられる．レファレンスサービスとは，利用者の質問や相談に応じて必要な資料や情報を提供するサービスである．レファレンス事例の一部はインターネット上で公開されているため，うろ覚えのデータとして利用できる．

本研究では，うろ覚えを許容する書籍の検索手法を開発することを目的とする．以下において，「うろ覚えの書籍タイトル」を「うろ覚え」，「正解書籍のタイトル」を「正解書籍」として記述する．

2. 従来システム・アルゴリズム

2.1 Google 検索

Google が提供する検索エンジンであり，セマンティック検索を導入し，入力があいまいでもユーザーの意図通りの結果を表示することが可能である[1]．

しかし，ジャンルが違うものなど直接関係ない検索結果が出力されることもあり，本を探したいときは検索語に「書籍」を含めるなど工夫が必要な場合も多い．本研究では，書籍のみに絞ることでこの課題の解決を目指す．

2.2 BM25

単語の出現頻度をもとにして順位付けを行う検索アルゴリズムの 1 つで，従来の手法である TF-IDF と比べ，文章の長さを考慮するように変更されている[2]．文章の長さを考慮することで，文章が長くマッチする単語が多いものよりも文章が短くマッチする単語が多いものを上位にでき，TF-IDF より精度が高いとされている．

‡秋田県立大学　Akita Prefectural University

しかし，文字でのスコア付けになっているため，うろ覚えで単語が類語に置き換わっている場合に検索できないことが考えられる．本研究では，単語のベクトル表現を用いて，ベクトル間の類似度を測ることで，単語の意味から検索し，この課題の解決を目指す．

2.3 Webcat Plus

国立情報学研究所が提供する書籍の検索サービスで、江戸時代から現代までに出版された書籍を対象に，それらの知識などを自由に探索できる思想空間の実現を目指し，現在は興味を持つ可能性のある本の提示を実現している[3]．日本全国書誌 JAPAN/MARC や BOOK データベース，大学図書館目録所在情報データベースなど様々なデータを本・作品・人物の軸で整理して提供している．一致検索と連想検索を提供しており，一致検索では検索結果の自動分類を行い関連性のある書籍の提示，連想検索では人間の思考のように例示によって関連情報を提供するなど，興味のある書籍をどこまでも探索できるようになっている．

本来想定されている使用方法ではないが，連想検索を使用すると本のあらすじから求めている本が出力されることも多い．しかし，連想を模した検索機能の特徴として，検索文中の単語があらすじやタイトルに含まれる書籍が上位となるため，検索文が短く正解書籍に含まれない単語が出現するうろ覚えからは，求めている書籍が提示されないことが多い．本研究では，単語のベクトル化を用いて，ベクトル間の類似度を測ることで，単語の意味から検索し，この課題の解決を目指す．

2.4 図書館が提供する蔵書検索

多くの図書館で提供されており，ほとんどの場合資料予約サービスと一体になっている．市立や県立の図書館では，図書館の検索機以外にインターネット上でも提供されており，図書館内外で利用できることが多い．

多くの図書館で提供されている蔵書検索は一致検索のみが多く，1 文字でも正解書籍に含まれない文字が含まれる検索文を入力すると，正解書籍が検索できない．一部の図書館では一致検索以外の蔵書検索も提供されており，簡単なうろ覚えであれば関連キーワードなどによって正解書籍が検索できることもあるが，うろ覚えと正解書籍がかなり違うものは検索することが難しい．本研究では，単語のベクトル化を用いてベクトル間の類似度を測ることで，単語の意味から検索し，この課題の解決を目指す．

3. 提案手法

3.1 提案手法の概要

本研究では，うろ覚えの書籍から求めている書籍を検索する手法を開発することを目的とする．提案するシステムは図 1 の構成になっており，入力はうろ覚えのみとし，類

似度の高い上位 10 件を出力とする．

データベースとして，書籍タイトルとそのベクトルを利用する．書籍タイトルのベクトルは，データベースの書籍タイトルモデルを学習し，そのモデルを使用して取得する．

システムは，うろ覚えを入力として受け付け，事前に作成したモデルに入力し，ベクトル化を行う．そのベクトルとデータベースのベクトルの類似度を計算し，その値の大きい上位 10 件を出力とする．

本提案手法では，モデルは fastText，類似度はコサイン類似度を使用する．

3.1.1 fastText

単語のベクトル表現を取得することができるモデル[4]で，未知語に対応できるという特徴がある．

fastText は単語ごとの n-gram であるサブワードを考慮することで活用形をまとめることが可能であり，サブワードを考慮するため，学習データに出現しない未知語にも対応できる．

学習アルゴリズムは skip-gram を使用する．学習の際に変更できるオプションには単語ベクトルの次元数，学習率，エポック数などがある．すべてデフォルトの値，単語ベクトルの次元数は 100，学習率は 0.1，エポック数は 5 を使用した．

書籍タイトルのベクトルは，書籍タイトルに含まれる単語ごとのベクトルをL^2ノルムで正規化しその平均を使用している．L^2ノルムとは，n次元ベクトル$\vec{x} = (x_1, x_2, \cdots, x_n)$があるとき，以下のように示すことができる．

$$L^2 = \sqrt[2]{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2}$$

3.1.2 コサイン類似度

2 つのベクトルがどのくらい似ているかを表す指標で，2 つのベクトルの大きさの差ではなく 2 つのベクトルがなす角の値であり，意味の近さで順位付けをすることができる．コサイン類似度の計算方法を以下に示す．

$$Cosine\ Similarity = cos(x, y) = \frac{\langle x, y \rangle}{\|x\|\|y\|}$$

3.1.3 学習データ・データベース

書籍のデータベースとして，断面日が 2022 年 6 月 14 日の国立国会図書館蔵書データを使用する．全蔵書データは約 1200 万件あり，評価実験に時間がかかるため絞り込みを行う．約 1200 万件から，NDC の分類番号を持ち，且つ，書籍タイトルの 1 文字目が記号・英語ではないもので絞り込んだ約 400 万件，そこに絞り込みからはじかれた評価データの正解書籍 175 件を加えた．

絞り込みに使用した NDC とは，日本十進分類法と言い，ジャンルごとの分類のことである．日本の多くの図書館で書籍の整理に使用されており，一般的な図書館に所蔵されている書籍データには含まれていると考え，国立国会図書館蔵書データ中の論文などを除くために使用した．また，英語書籍や 1 文字目が記号の書籍はレファレンスで聞かれることが少ないと考え，絞り込みに使用した．記号と英語の判定には形態素解析エンジンの Mecab を使用した．

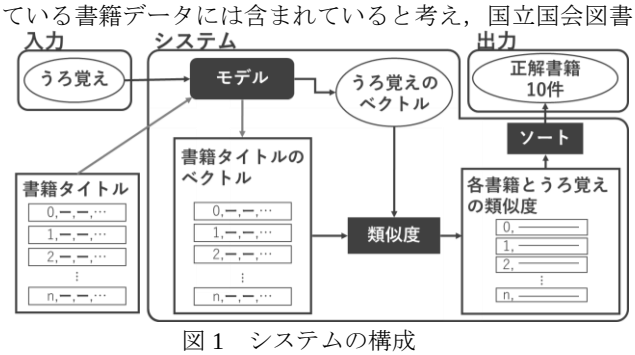

図 1　システムの構成

fastText モデルの学習には書籍タイトルデータのみを使用するため，絞り込みを行った書籍タイトルと書籍タイトルの分かち書きを学習データとして用いる．分かち書きには Mecab を使用し，辞書は Unidic を使用した．

4. 評価実験

4.1 評価方法

入力をうろ覚え，出力の上位 10 件に正解書籍が含まれるかで評価を行う．提案手法で埋め込みに使用するモデルは fastText，比較手法は従来システムの Google 検索と BM25 とする．

4.1.1 評価データ

評価データとして，福井県立図書館覚え違いタイトル集[5]とレファレンス協同データベース[6]のレファレンス事例を使用する．

福井県立図書館覚え違いタイトル集では， 2022 年 5 月時点での事例 1057 件から，覚え違いではない例を除き，2 つ以上のうろ覚えがまとめられたデータをうろ覚え 1 件ごとに分割した 1064 件を用いる．レファレンス協同データベースでは，2022 年 9 月時点でのレファレンス事例のうち，解決/未解決を解決，内容種別をうろ覚え・覚え間違いで絞り込んだ 299 件を用いる．

2 つのデータ計 1363 件に処理を行う．まず，タイトルとみなすことができるデータが含まれる 1058 件に絞りこむ．また，うろ覚えと正解書籍のみを抽出し，正解書籍が部分一致する国立国会図書館蔵書データを抜き出し，正解として書籍データが複数抜き出せたものから，正解書籍としてデータを 1 件抽出する．抜き出せた書籍データが 1 件のものと複数書籍から 1 件抽出できたデータ計 852 件を今回の評価データとして使用する．

4.2 結果

各手法の上位 10 件正解率を表 1 に示す．Google 検索では結果しか算出できずスコアは表示されないため，提案手法と BM25 の結果を図示する．評価データを 50 件サンプリングし，その上位 10 件のコサイン類似度の箱ひげ図と正解のスコアの散布図を提案手法，BM25，それぞれ図 2 と図 3 に示す．サンプルはランダムに取得し，提案手法と BM25 は同じ評価データである．すべてのデータのスコアと，うろ覚えと正解書籍の文字列の長さの差との関係を提案手法と BM25，それぞれ図 4，図 5 に示す．うろ覚えと正解書籍の文字列の長さの差の計算方法は以下のように示すことができる．

文字列の長さの差 = うろ覚えタイトルの長さ
− 正解書籍のタイトルの長さ

表 1 上位 10 件の正解率

手法	正解率[%]
提案手法	30.1
Google 検索	49.6
BM25	43.2

そのため，正解書籍のタイトルがうろ覚えタイトルより長いと負の数となり，うろ覚えが正解書籍のより長いと正の数となる．

4.3 考察

表 1 から，Google 検索の正解率が 1 番高く，提案手法は従来手法に比べ正解率が低い．図 2 と図 4 を比較すると，正解書籍のスコアと上位 10 件の距離は提案手法より BM25 のほうが近いため，BM25 は，不正解の場合であっても正解書籍のスコアは上位 10 件に近いと考えられる．図 3 と図 5 から，正解書籍よりうろ覚えの方が長い文字列のデータが多いことが分かる．また，図 3 と図 5 を比較すると，提案手法では文字列の長さの差が小さいものしか正解を導くことができておらず，正解書籍のコサイン類似度が異様に低く算出されているデータが多い．また，実際のデータを分類してみると，うろ覚えのデータの半数近くは意味が近い単語に置き換わっているが，提案手法は BM25 と比べてその項目の正解率が高いとは言えない．約 400 万件の書籍タイトルを学習に使用しており，学習する単語は膨大な数だと考えられるが，100 次元で学習を行っているため，意味の情報がうまく取得できていない可能性がある．従来手法 2 つと比べ，正解書籍にサブタイトルやタイトルの英訳が含まれている場合の正解率が著しく低い．提案手法では文全体のコサイン類似度で算出しているため，サブタイトル等がノイズとなっている可能性がある．

5. 追加実験

文全体のコサイン類似度では，サブタイトルがかなりのノイズになるなど，正解書籍とうろ覚えで一致している単語や意味的に近い単語自体を重視することができていない．そのため，文全体のコサイン類似度以外の類似度との比較を行う．

5.1 評価方法

3.1 の構成と同じく，類誌度だけを変更して同様に評価を行う．ベースラインとして BM25，Google 検索，比較する類似度としてコサイン類似度，Word Mover’s Distance，Word Rotator’s Distance，Word Mover’s Distance と Word Rotator’s Distance を組み合わせた手法の 4 種類を使用する．

モデルは先ほどと同じものを使用し，データベースはうろ覚えの正解を含む 1 万件のデータベースを計 5 回作成し，正解率の平均を計算する．評価データは新たに絞り込みを行ったレファレンス事例のデータを使用する．

5.1.1 評価データ

評価データとして，同じく福井県立図書館覚え違いタイトル集とレファレンス協同データベースのレファレンス事例を使用する．絞り込みの途中の 1058 件のうち，読み仮名のデータを持つ書籍タイトルで絞り込んだ 913 件を使用する．

5.1.2 データベース

うろ覚えに対する正解書籍を 1 件以上含む 1 万件のデータベースを 5 セット作成した．まず，先ほどと同様の約

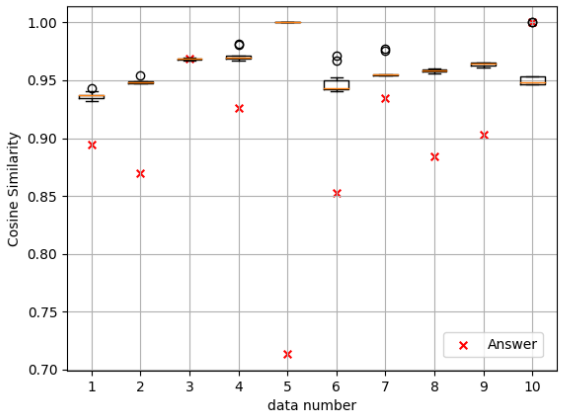

図 2 提案手法:上位 10 件と正解のコサイン類似度

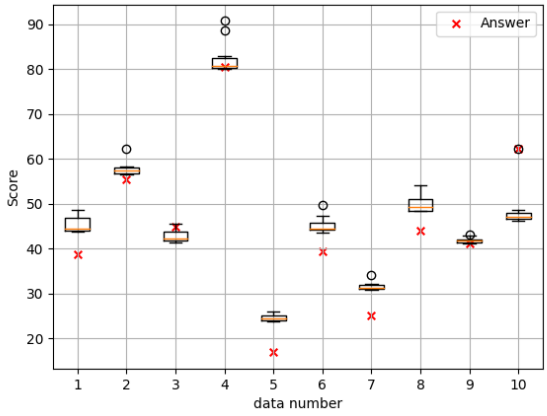

図 3 BM25:上位 10 件と正解のスコア

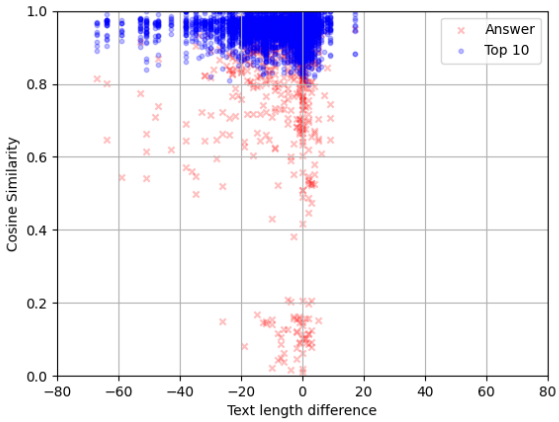

図 4 提案手法:文字列の長さの差とコサイン類似度

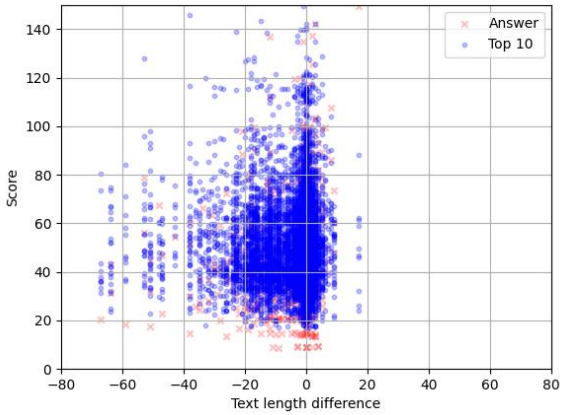

図 5 BM25:文字列の長さの差とスコア

1200 万件の書籍データから，書籍タイトルと書籍タイトルの読み仮名をもつデータに絞り込む．そこから，正解データを取得し，正解データと重複しないよう，絞り込んだデータから 1 万件になるようにデータを取得する．これを 5 回行った．

5.2 新たに使用する類似度

5.2.1 Word Mover's Distance (WMD) [7]

文間距離を計算するアルゴリズムの 1 つであり，文を一様な重みを持つ離散分布ととらえ，ユークリッド距離をコストとし，最適輸送を使用して距離を計算する．

Twitter などの短文のタスクにおいて高い精度を示しており，単語が入れ替わっていても対応できることから，特定の単語を重視することができるのではないかと考え，使用する．

5.2.2 Word Rotator's Distance (WRD) [8]

WMD を改良した，文間距離を計算するアルゴリズムの 1 つで，WMD では， 単語の重要度と意味を取り混ぜてしまっており，単語の重要度はベクトルのノルム，単語の意味はコサイン類似度で計算できることから，文をノルム分の重みを持つ離散分布ととらえ，コサイン類似度をコストとして，最適輸送を使用して距離を計算する．

単語の入れ替わりにも対応でき，コサイン類似度を用いているため，単語の意味にも対応できると考え，使用する．

5.2.3 WMD と WRD を組み合わせた手法 (WM_RD)

WMD では，横井ら[8]が指摘した通り，単語ベクトルの意味的類似度はコサイン類似度で測ることができるのにも関わらず，WMD では重要度と意味が混ざったユークリッド距離で計算している．そのため，距離尺度はコサインで計算するのが望ましいと考える．しかし，今回の場合，単語の重要度が高い単語がうろ覚え中で重要視したい単語というわけではない．そのため，WMD のように単語ベクトルに一様な重みをもたせた離散分布とし，コサイン類似度をコストとして最適輸送を使用して距離を計算する．

5.3 結果と考察

各手法の正解率を表 2 に示す．5 回計算しているが，どの手法もばらつきは大きくなかった．WMD の正解率が一番高く，次にコサイン類似度が BM25 より低いのは WRD のみとなった．

WRD よりも WM_RD の正解率のほうが高いため，うろ覚えを検索する場合には，一様な離散分布ととらえるほうが良いことが分かった．WRD のみが正解したデータを見てみると， 他の手法のみが正解しているものより語感が近いものや文字種が異なるもので正解していることがわかった．単語の意味と重要度を取り混ぜたユークリッド距離を使用することで，うろ覚えの不安定さを吸収できているのではないかと考える．コサイン類似度が正解したデータを見てみると，400 万件のデータベースでは正解していなかったものも正解しているため，データ数が減ったことで類似度が高いデータが少なくなったためだと考えられる．

表 2 1 万件データベースでの上位 10 件の正解率

	1	2	3	4	5	Average
BM25	40.0	40.2	40.1	39.8	39.2	**39.8**
Google						**48.5**
COS	50.7	51.2	50.1	50.4	50.6	**50.6**
WMD	59.3	59.0	60.0	59.5	58.7	**59.3**
WRD	39.2	38.9	39.4	39.2	39.5	**39.3**
WM_RD	46.3	46.4	45.9	46.8	46.5	**46.4**

6. おわりに

ジャンルが限定されたうろ覚えの検索では，既存の検索システムでは不十分なことが多く，検索に時間がかかることや検索できないことが考えられる．うろ覚えと正解のデータとして，事例が公開されている図書館のレファレンスサービスに注目し，本研究では，タイトルのうろ覚えを許容する書籍検索の手法を開発することを目的とした．

提案手法では，fastText モデルによるベクトル化とコサイン類似度による順位付けを行った．結果，提案手法の正解率は従来手法である Google 検索と BM25 を大きく下回った．うろ覚えのデータの半数近くは意味が近い単語に置き換わっているが，その項目の正解率が高いとは言えないため，単語の意味を学習できていないと考えられる．今後の課題である提案手法の改善のために，システムの変更として，fastText モデルのベクトルの次元数の増加や，BERT などのモデルの利用．学習データの変更や組み合わせ手法として，入力誤りデータ，読み仮名データ，Wikipedia のダンプデータの利用などを検討する．

また，追加実験では，1 万件のデータベースを使用して，BM25 と文全体のコサイン類似度，WMD，WRD，WMD と WRD を組み合わせた手法の正解率を比較した．WMD の正解率が一番高いという結果だった．コサイン類似度の正解率が 400 万件のデータベースを使用した場合よりも 20%程度高いため，大規模なデータベースでも確認したい．

謝辞

本研究では国立国会図書館で作成された国立国会図書館サーチ書誌全件断面ファイル，福井県立図書館で作成された覚え違いタイトル集，全国の図書館から作成されたレファレンス協同データベースのデータを利用させていただきました．ここに感謝の意を表します．

参考文献

[1] Google 検索-概要 (アクセス日 2023/06/14) https://www.google.com/intl/ja/search/howsearchworks/
[2] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford, Okapi at TREC-3, TREC-3, pp.109 (1994)
[3] Webcat Plus (アクセス日 2023/06/14) http://webcatplus.nii.ac.jp/#
[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, Enriching Word Vectors with Subword Information, arXiv (2016)
[5] 福井県立図書館覚え違いタイトル集 (アクセス日 2022/05/30) http://www.library-archives.pref.fukui.lg.jp/tosyo/category/shiraberu/368.html
[6] レファレンス共同データベース (アクセス日 2022/09/22) https://crd.ndl.go.jp/reference/
[7] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger, From Word Embeddings To Document Distances, in: Proceedings of the 32nd International Conference on Machine Learning, PMLR 37, pp.957-966 (2015)
[8] 横井祥，高橋諒，赤間怜奈，鈴木潤，乾健太郎，超球面上での最適輸送に基づく文類似性尺度，言語処理学会，第 26 回年次大会，pp.335-338 (2019)