E-013

レファレンスサービス自動化のための書籍分散表現

松橋 勇輝†　　安達由洋††
†イー・アンド・エム株式会社
††東洋大学総合情報学部総合情報学科 adachi@toyo.jp

要約

現在、図書館業界は従来の資料貸出を中心としたサービスから情報提供サービスの展開へと方針付けられている。しかし、生涯学習社会の普及により求められる情報が高度化し、一方で指定管理者制度により司書が減少することで高度な情報サービスを人手のみで提供することは難しくなりつつある。そこで、近年急速に発達している情報技術を用いて情報サービス提供を支援あるいは自動化することが望まれる。

本研究では、情報サービスの中でも中心的なレファレンスサービスの核となる書籍検索の支援・自動化を念頭に、書籍分散表現による検索手法に関する二つの実験を行った。一つ目の実験では、レファレンスサービスの自動化にどのような分散表現手法が適しているかを調査した。実際に書籍の本文テキストから分散表現を得て、質問文との類似度が最も高い上位 5 冊の書籍を取り出す実験を行った。これを様々な分散表現手法で行い、検索結果を比較・評価した結果、non-SCDV が最も高い評価値を獲得した。二つ目の実験では、書籍を分散表現に基づいて内容のまとまりごとに分割する方法を提案するとともに、分割された文書を検索対象として non-SCDV を用いた書籍検索実験を行うことで検索精度の変化を調査した。実験の結果、書籍分割を用いた検索が分割してない書籍に対する検索を上回る評価値を獲得した。以上の実験により、レファレンスサービス自動化のための核となる書籍検索機能の実現に non-SCDV に基づく書籍分散表現が有効であり、かつ独自に提案した書籍分割手法がより適した手法であることが検証された。

1. はじめに

図書館はこれまで、資料の閲覧・貸出サービスを重視してきた。この方針は「中小都市における公共図書館の運営[1]」を反映したもので、図書館を日本に普及させることを目的としていた。この方針が功を奏し、図書館は広く日本に普及した。一方で、閲覧・貸出サービスばかりが注目されることで、図書館が無料の貸本屋であるという誤解と他のサービスの軽視を招いた。この態勢は近年大きく見直され始めている。特に、図書館は「これからの図書館の在り方検討協力者会議[2]」を受け、これまでの閲覧・貸出を中心としたサービスから、情報提供を中心としたサービスの展開へと方針付けられた。一方、利用者の求める情報は、生涯学習社会・高学歴社会化に伴い多様化・高度化している。このような需要に対し、図書館はより広範な資料の収集と前述の情報サービスの拡充を同時に行うことが求められている。しかし、全国的な予算の縮小や指定管理者制度による運営など、図書館業界を取り巻く現状ではこの需要を満たしうる情報サービスの提供は難しい。そこで、この情報サービス提供に関する問題を近年急速に発達している情報技術を用いて支援あるいは自動化することが望まれている。

本研究では情報サービスの中でも中心的なレファレンスサービスの核となる書籍の検索の支援・自動化を、自然言語の分散表現を用いて取り組む。レファレンスサービスとは、「何らかの情報を求める利用者の質問（参考質問）に対して、回答となる情報そのものを提供したり、回答の含まれる情報源を指示・提供すること[3]」である。

レファレンスサービスを自動化する試みはこれまでにも行われてきた[4]。本研究では、従来使われていない自然言語の分散表現を利用したレファレンスサービスの自動化に取り組む。分散表現に関する関連研究として、分散表現を利用した論文分類・検索がある[5][6][7]。これらの研究で扱う学術論文・特許は比較的短く単一の主題からなる文書である。一方で、書籍や雑誌などは複数の主題からなる長い文書であり、レファレンスサービスにおいては比較的短い質問文と文書とを結びつけなければならない。この点において、レファレンスサービスを自動化するには単に既存の論文検索手法を適応するだけでは難しい

と思われる。

本研究では、レファレンスサービスの自動化に向けて分散表現を用いた書籍検索に関する二つの実験を行った。一つ目の実験は、様々な分散表現手法で書籍の本文テキストの分散表現を求め、与えられた質問文の分散表現との類似度を算出する。類似度が最も高い上位五冊を取り出し、それらの書籍に質問の解答となる情報が含まれているかを判定する。この結果をもとに、レファレンスサービスの支援・自動化システムに適した分散表現手法について比較・検討した。用いた分散表現手法は Skip-gram[8]、CBOW[8]、PV-DM[9]、PV-DBOW[9]、TF-IDF、SCDV[10]、non-SCDV[10]である。

二つ目の実験では、書籍の本文テキストを分散表現に基づいて内容のまとまりごとに分割する手法を提案するとともに、この手法に基づいて分割した文書を検索対象とすることで質問文と合致する書籍を検索する性能が向上するかを調査する。

本研究の貢献を以下に示す。

- 書籍の様々な分散表現手法による分散表現を求め、質問文の分散表現とのコサイン類似度を用いた検索結果を評価・比較した。
- 書籍を分散表現に基づいて内容のまとまりごとに分割するという独自の手法を提案し、分割した書籍分散表現と書籍全体を分散表現する従来手法とで検索結果を比較した。

2. 書籍分散表現による検索

本節では、様々な分散表現手法を用いて質問文と合致する書籍を出力する実験について述べる。

2.1. 分散表現手法

実験に用いる各分散表現（ベクトル化）手法について概説する。

A. Skip-gram / CBOW

Skip-gram 及び CBOW（Continuous Bag of Words）は、Tomas Mikolov らが提案した単語をベクトルで表現する手法である[8]。3 層のニューラルネットワークからなり、ある単語の前後に出現する他の語からベクトル化する。CBOW は、周囲の単語から対象となる単語を予測するモデルである。Skip-gram は、ある単語からその周囲に出現する単語を予測するモデルである。

本実験では、Skip-gram および CBOW を用いて単語ベクトルを作成する。文書ベクトルは文書内に出現する全ての単語のベクトル表現を足し合わせて平均をとったものを用いる。

両モデルのパラメータとして、次元数は 200、前後の考慮する単語の数は 5、単語を考慮する最低出現回数は 1 と設定する。なお、実装は gensim 社の Word2Vec を用いた。明示していないパラメータはデフォルトの値とする。

B. PV-DM / PV-DBOW

PV-DM 及び PV-DBOW は、Quoc Le 等[9]が提案した文書をベクトルで表現する手法である。Skip-gram 及び CBOW を元に発展させ、ベクトル化の対象を文書に拡張したものである。PV-DM は CBOW を応用したもので、CBOW の入力ベクトルに単語列とともにパラグラフ ID を与えたものである。PV-DBOW は Skip-gram を応用したもので、Skip-gram の入力を単語ではなくパラグラフ ID にしたものである。

本研究では、両モデルのパラメータとして、Skip-gram / CBOW と同様に次元数を 200、前後の単語数を 5、単語の最低出現回数を 1 と設定する。なお、実装は gensim 社の Doc2Vec を用いた。明示していないパラメータはデフォルトの値とする。

C. TF-IDF

TF-IDF とは、文書中に含まれる単語の重要度を測る指標である。ある単語の出現頻度を表す TF（Term Frequency）と、ある単語の逆文書頻度である IDF（Inverse Document Frequency）の積からなる。ある単語について、ある文書内での出現頻度に全文書における出現頻度の逆数を掛けることにより、ある単語がその文書に特徴的な語である度合いを測ることができる。

本実験では、文書ベクトルは全文書に出現する単語の並びとし、各単語の TF-IDF を要素とする。

このモデルにおける TF-IDF では、アルファベットの大文字と小文字は区別している。また、IDF に 1 を加算している。この処理により、全ての文書に登場することで TF-IDF が 0 となり結果に影響を与えない単語であっても結果に影響を与えることができるようになる。なお、実験では TF-IDF の計算に scikit-learn を用いた。明示していないパラメータはデフォルトの値とする。

D. SCDV

SCDV（Sparse Composite Document Vector）は、既存の単語のベクトルを補正し、より精度の高い分散表現を獲得する手法である[10]。

ある文書において、任意の手法によって単語の分散表現を取得する。これらのベクトルをクラスタリングし、ク

ラスタごとに各クラスタのいずれかに属するかの確率を求める。これらのベクトルと各クラスタに属する確率の積を連結し一つのベクトルとしてまとめる。このベクトルとこのベクトルが表現している単語の IDF の積を算出する。得られたベクトルを word-topics vector という。SCDV は、対象の文書に出現する単語の word-topics vector を足し合わせていき、スパース化することで取得する。

本研究では SCDV を用いたモデルを条件別に 3 つ用いる。SCDV は、元となる単語の分散表現が必要となる。本実験では Skip-gram と CBOW を用意した。また、SCDV には特徴の鮮明化等のためスパース化を行う。一方で、スパース化をすることで僅かな値の特徴が無視されることとなる。本実験では、無視される僅かな特徴が出力結果に影響を及ぼす可能性を考慮し、スパース化しないモデル non-SCDV[10]も用意した。このモデルでは単語の分散表現に Skip-gram を用いる。本実験では、以上の 3 つのモデルを用いる。

本実験で用意した Skip-gram と CBOW のパラメータは前述と同様に次元数 200、前後の単語数 5、単語の最低出現回数 1 である。ベクトルをクラスタリングする手法としてガウス混合モデル（以下、GMM と呼称する）を用いた。GMM のパラメータとして、クラスタ数を 30、最大繰り返し回数を 50 とした。ベクトルを修正するための IDF は前述と同様に、アルファベットの大文字と小文字を区別し、1 を加算する。また、Skip-gram の実装は gensim 社の Word2Vec を、GMM の実装は scikit-learn を用いた。明示していないパラメータはデフォルトの値とする。また、スパース化の閾値パラメータは 4 を用いている。

2.2. 分散表現による検索実験

書籍の本文テキストを上述の各分散表現手法でそれぞれ分散表現し、与えられた質問文の分散表現とのコサイン類似度を算出する。類似度が最も高い上位 5 冊を取り出し、それらの書籍に質問の解答となる情報が含まれているかを判定する。質問文は実際のレファレンスサービスの事例を元に 5 つ用意した。

質問文の例を以下に示す。

・ バンプマッピングが活用できる場面を知りたい
・ シュメール人の食事について調べたい

2.3. データセット

実験に使用するデータセットは、実際の書籍・雑誌 670 冊をテキストデータ化することで収集した。内訳は書籍 279 冊、逐次刊行物 391 冊である。書籍のうち、学術に関するものは 259 冊、文芸ほか 20 冊である。逐次刊行物のうち、学術に関するものは 390 冊、文芸ほか 1 冊である。

テキストデータは以下の前処理を行った。

(1) 記号等の除去
　本文の内容に関係ない記号等が含まれるため除去する
(2) 分かち書き・形態素解析
　テキストを分かち書き・形態素解析する
　形態素解析エンジンは MeCab[11]を用いる
(3) 特定の品詞の抽出
　文書の内容に影響する品詞のみを抽出する
　本実験では名詞、動詞、形容詞を抽出した
(4) 基本形への変換
　動詞、形容詞を基本形に変換する
　活用による表記の違いをなくすため

以上の前処理を行った書籍データおよび質問文をもとに前述の各分散表現手法を用いた分散表現を生成する。

2.4. 分散表現を用いた検索結果

本実験では、評価指標として以下の評価式を用いる。この評価式は各質問に対する出力結果である五冊の書籍が妥当かどうかを評価するためのものである。以下のうち、N は出力される書籍の冊数を表す。本実験では五冊の書籍を出力するので N は 5 となる。i $(i \in \{1, 2, \cdots, N\})$はその書籍が何番目に出力されたかを表す。最も類似度が高い書籍が一番目に出力されるので i が 1、次が 2、3 と続く。n_i $(n_i \in \{0, 0.5, 1\})$はその書籍が質問の解答としてどの程度妥当かを表す。出力された書籍の内容に、質問文の直接的な解答となる記述があるとき、その書籍の n_iを 1 とする。解答となる記述は含まれていないものの、質問と同一の主題が述べられている書籍の場合、n_iは 0.5 とする。質問と書籍に関連がない場合、n_iは 0 とする。

本実験では五つの質問を与えている。各質問に対して評価値（score）を以下の式で算出し、その平均をとることで各分散表現手法の評価値とした。

$$score = \sum_{i=1}^{N} n_i \times (N + 1 - i)$$

$$n_i = \begin{cases} 1 \\ 0.5 \\ 0 \end{cases}$$

表 1 に各分散表現を用いた書籍検索実験の結果を示す。

表 1. 各分散表現手法の評価値

	Q1	Q2	Q3	Q4	Q5	平均
Skip-gram	11.5	10	6	11.5	13.5	10.5
CBOW	11	4	3	12	12.5	8.5
PV-DM	7.5	5	0	11	7	6.5
PV-DBOW	6	1	0	5	13	5
TF-IDF	12	10	5	7.5	10	8.9
SCDV (Skip-gram)	8	12	12	9.5	14	11.1
SCDV (CBOW)	11	13	8.5	8	12.5	10.6
non-SCDV	12	10	12	11	13	11.6

表 1 より、最も評価値の高い手法は non-SCDV である。続いて SCDV（Skip-gram）、SCDV（CBOW）の順に高い評価値を得た。

2.5. 分散表現を用いた実験の考察

評価値の上位 3 モデルが全て SCDV となった理由として、語の属するカテゴリと同一カテゴリの語の類似度が高くなる SCDV 特徴が質問文と書籍の本文テキストとの類似度を算出するうえで効力を発揮したものと思われる。

non-SCDV が最も高い評価値となった理由として、質問文テキストという短いテキストと書籍本文テキストという長いテキストとの類似度を算出するという本実験の設定では、スパース化によって無視される僅かな特徴が検索情報として有用であったと考えられる。

以上のことから、non-SCDV を用いた分散表現による書籍検索がレファレンスサービスの自動化に適した手法だといえる。

3. 書籍分割を用いた検索

本節では、書籍を内容のまとまりごとに分割し、それを検索対象として 2 節と同様の実験を行い、評価値がどのように変化するかを調査した。

3.1. 文書分割アルゴリズム

前述したように、書籍や雑誌などは複数の主題からなる長い文書であり、レファレンスサービスにおいては比較的短い質問文と文書の対応する部分をうまく結びつけなければならない。このために、分散表現に基づいて文書を同一の主題を扱う部分ごとに分割する文書分割アルゴリズムを提案する。

入力するデータは文書を構成する単語系列$W = \{w_1, w_2, \cdots, w_T\}$である。系列の先頭から一定の単語数の部分系列$W_{1,i}$を取り出す。更に、$W_{1,i}$の終端位置に続いて同じ長さのテキスト$W_{i+1,2i}$を取り出す。次に、$W_{1,i}$と$W_{i+1,2i}$の分散表現をそれぞれ求め、それらの類似度を算出する。類似度が閾値以上であれば、$W_{1,i}$と$W_{i+1,2i}$は同じ主題について記述したテキストと判定する。類似度が閾値未満であれば、$W_{1,i}$と$W_{i+1,2i}$の間で主題が変化したと判定し、$W_{1,i}$を一つの主題を記述したテキストとして切り出して残りの部分を単語系列Wとする。類似度が一定以上の場合は、取り出すテキストの単語数を増加させて再度同じ処理を行う。これを系列の最後まで行う。本論文末の図 1 に文書分割アルゴリズムを示す。

この文書分割アルゴリズムには四つのパラメータがある。一つは取り出す部分単語系列の初期の長さ（init_size）である。この値は部分単語系列が記述している主題を表現することができる最小の単語数に設定する。二つ目のパラメータは、部分単語系列の長さの増分（increment）である。部分単語系列間で主題が変化していないと判定した場合、長さを変化させてもう一度分散表現化・類似度算出を行う。このとき、部分単語系列の長さをどの程度増やすかを increment で設定する。この値は小さいほど詳細な文書分割を行うことができるが、計算時間が増大する。三つ目のパラメータは部分単語系列間の主題が変化したかを判定する類似度の閾値（threshold）である。類似度がこの値以上であれば部分単語系列間の主題は変化していないと判定し、この値未満であれば主題は変化したと判定する。この値は小さいほど主題の変化に鈍感になり、大きいほど敏感になる。すなわち、値が小さいほど詳細に文書を分割し、値が大きいほど粗掴みに分割する。四つ目のパラメータは分散表現を求める部分単語系列の長さの上限値（max_size）である。この文書分割アルゴリズムでは、部分単語系列の長さが非常に大きくなると文書全体の分散表現ベクトルに近づき、主題の変化に対して鈍感になる。これを防止して適切な文書分割を行わせるために分散表現ベクトルを求める部分単語系列の長さを max_size 以下に制限する。

3.2. 書籍分割を用いた検索実験

本実験では、文書分割アルゴリズムを用いて書籍の本文テキストを分割し、それを検索対象として 2 節と同様の実験を行った。

2 節の実験で使用した 670 冊の書籍・雑誌を 2.3 節で述べた方法で前処理したのち、図 1 の文書分割アルゴリズムで分割した。これにより得られた書籍断片 3441 個を検索対象として質問文との類似度比較を行った。

3.3. 書籍分割での使用モデル

本実験で使用した文書分割アルゴリズムのパラメータは、初期の部分テキストの単語数 init_size を 100、単語数の増分 increment を 10、主題が変化しているかを判定する閾値 threshold を 0.3、分散表現を求める部分単語系列の長さの上限値 max_size を 300 とした。また、文書分割に用いる分散表現手法として non-SCDV を用いた。

また、質問文も 2.3 節で述べた前処理をしたのち non-SCDV により分散表現化した。non-SCDV の各パラメータは 2 節で使用したものと同様である。

3.4. 書籍分割を用いた検索結果

表 2 に書籍分割を用いた検索実験の結果を示す。

表 2. 独自手法の評価値

Q1	Q2	Q3	Q4	Q5	平均
11	15	10	13	13.5	12.5

表 2 より、分散表現に基づく書籍分割を用いた検索結果は 2 節で述べた分割しない検索結果よりも高い評価値を得た。

3.5. 分割表現を用いた実験の考察

本実験の書籍分割に基づく検索手法と 2 節の分割しない検索手法の違いは、書籍を同じ主題のまとまりごとに分割する処理を事前に行っているかどうかだけである。それ以外の non-SCDV を用いた分散表現・類似度算出等の処理に違いはない。つまり、本研究で提案している分散表現に基づく書籍の主題ごとへの分割は、一般に複数の主題を含む書籍の検索に有効である事を示している。

一方で、この手法には幾つかの問題点がある。一つは、検索処理時間の増大である。本手法では、書籍の本文テキストを分割したものを検索対象としているため、分割しない検索手法より多くのベクトル間類似度の計算が必要となる。分割されるテキストの数はパラメータにより変化するが、本実験では 670 冊の書籍テキストが 3441 個の断片に分割された。これにより、一度の検索に必要な類似度計算の回数が増加するため、検索の応答時間が遅くなる。

また、分散表現に基づいた書籍分割には、何度も文書の分散表現化と類似度計算を行う必要がある。多数の書籍を検索対象とする場合には、この計算時間が問題となる。

4. 結論

レファレンスサービスの自動化のため不可欠な書籍検索に関する二つの実験を行った。一つは、様々な分散表現手法を用いて質問文から適切な書籍を検索する実験である。その結果、non-SCDV を用いた分散表現が最も高い評価値を得た。もう一つは、書籍を分散表現に基づいて内容のまとまりごとに分割する独自の手法を用いて書籍検索実験を行った。その結果、この書籍分割に基づく検索手法が最も高い評価値を得る事を確認した。このことにより、分散表現に基づく書籍分割とそれを用いた書籍検索がレファレンスサービスの自動化に有効な手法であることが示された。

本研究で提案した書籍分割により分割された文書断片に元の書籍の対応する章やページ情報などをタグ付けしておくと、質問文の内容がどの書籍のどの章に書かれているか、どのページに書かれているかなどの情報を提供することができる。このような機能もレファレンスサービスとして重要になると思われる。

本研究では、小規模の書籍データと質問文で検索実験を行ったが、今後より大規模なデータセットで実験を行うことが課題である。

文献

[1] 日本図書館協会，“中小都市における共図書館の運営—中小公共図書館運営基準委員会報告”，日本図書館協会，1973.

[2] 文部科学省，“これからの図書館の在り方検討協力者会議”， http://www.mext.go.jp/b_menu/shingi/chousa/shougai/019/index.htm，2019 年 1 月 30 日アクセス.

[3] 日本図書館協会用語委員会，“図書館用語集 四訂版”，公益社団法人 日本図書館協会，2015.

[4] 黒橋禎夫，日笠亘，“京都大学附属図書館における自動レファレンス・サービス・システム”，https://www.jstage.jst.go.jp/article/johokanri/44/3/44_3_184/_pdf

[5] 田中昌昭，“単語の分散表現を用いた文書分類”，川崎医療福祉学会誌，Vol.28，No.1，pp.167-178，2018.

[6] 小林雄太，松本裕治，“論文の構成要素を考慮した分散表現に基づく類似論文検索”，言語処理学会 第 24 回年次大会 発表論文集，pp.959-962，2018.

[7] 安藤俊幸，“機械学習を用いた効率的な特許調査方法”，Japio YEAR BOOK 2018，pp.238-249，2018.

[8] Mikolov T., Chen K., Corrado G. and Dean J., “Efficient Estimation of Word Representations in Vector Space”, https://arxiv.org/pdf/1301.3781v3.pdf, 2013.

[9] Le Q. and Mikolov T., “Distributed Representations of Sentences and Documents”, https://arxiv.org/pdf/1405.4053.pdf, 2014.
[10] Mekala D., Gupta V., Paranjape B., and Karnick H., “SCDV : Sparse Composite Document Vectors using soft clustering over distributional representations”, https://arxiv.org/pdf/1612.06778.pdf, 2017.
[11] “MeCab: Yet Another Part-of-Speech and Morphological Analyzer”, http://taku910.github.io/mecab/, 2019 年 2 月 24 日アクセス.

Algorithm: 文書分割
Data: $W = [w_0, w_1, \cdots, w_{T-1}]$ #単語系列
Result: D #タプル（部分単語系列, 分散表現ベクトル）の系列
init_size #初期単語系列の長さ
max_size #分散表現ベクトル化を行う単語系列の長さの上限値
threshold #類似度の閾値
D = [ ]
count = 0 #単語長の増分カウント
start_index = 0
end_index = init_size - 1
while True:
    if end_index < $T - 1$:
        blockA = W[start_index : end_index +1]
        if end_index + init_size + count * increment < $T - 1$:
            blockB = W[end_index + 1 : end_index + init_size + count * increment +1]
        else:
            blockB = W[end_index + 1 :]
        if len(blockA) <= max_size:
            vectorA = 分散表現ベクトル化(blockA)
        else:
            vectorA = 分散表現ベクトル化(W[end_index - max_size + 1 : end_index +1])
        if len(blockB) <= max_size:
            vectorB = 分散表現ベクトル化(blockB)
        else:
            vectorB = 分散表現ベクトル化(W[end_index + 1 : end_index + max_size +1])
        cos_sim = コサイン類似度を計算(vectorA, vectorB)
        if cos_sim < threshold:
            D.append((blockA, vectorA))
            start_index = end_index + 1
            end_index = start_index + init_size
            count = 0
        else:
            count++
            end_index = end_index + increment
    else:
        block = W[start_index :]
        vector = 分散表現ベクトル化(block)
        D.append((block, vector))

図 1. 文書分割アルゴリズム