Vol.2021-IFAT-142 No.5
Vol.2021-DC-120 No.5
2021/3/26

文書クラスタリングへの文脈付きで非対称な単語類似度の応用

岸田 和明[1,a)]　門脇 夏紀[2]

概要：Twitter におけるツイートに代表される，いわゆる「短いテキスト」に対して，文書クラスタリングの手法を適用する際には，自動的に語を追加するなどして，その表現を補っておくことが望ましい。これは，この種の短いテキストには十分な数の語が含まれないことから，同義語などの表記のゆれが，悪影響を及ぼす可能性が高いためである。本研究では，この問題に対してクエリ拡張（query expansion）の手法を適用する。その際の語間の類似度の計算には，従来的な類似度シソーラスのほか，翻訳確率の推定法である IBM Model 1 に基づく独自の手法を使う。これは，潜在ディリクレ配分（LDA）により検出されたトピックを文脈とする非対称の類似度である。

1. はじめに

Twitter におけるツイートに代表される，いわゆる短いテキスト（short text）に対して，機械学習やクラスタリングの手法を適用する際には，自動的に語を追加するなどして，その表現を補っておくことが望ましい。これは，この種の短いテキストには十分な数の語が含まれないことから，同義語などの表記のゆれが，特に悪影響を及ぼす可能性が高いためである。

例えば，クラスタリングの対象となる 2 つの文書がそれぞれ，「automobile, vehicle」「motorcar, vehicle」という語を含んでいた場合，「vehicle」によって，これらは同一のクラスタにまとまるかもしれない。それに対して，単に「automobile」「motorcar」だけが含まれるならば（その他の語も共有されないとして），文書間の類似度は自動的にゼロとなり，同様な概念を扱っているにもかかわらず 1 つのクラスタにはまとまらない可能性がある。このようなケースは短いテキストの場合に生じやすい。

その解決策としては，

(1) 語の分散表現を使って，文書を CBOW(continuous bag-of-words）によるベクトルで表現する
(2) 語間の類似度に基づいて，元の文書に語を追加する

ことが考えられる。(1) の方法については，Curiskis ら (2020)[2] や門脇 (2020)[5] が Word2Vec を使った文書クラスタリングの実験を試みている（前者のデータは SNS，後者は Reuters の記事）。現在の研究状況では (1) の方法が有望であるものの，本研究では，語間の類似度による (2) の方法に焦点を当てることにする。

上記の例の場合，短いテキストの集合に対するクラスタリングの前に，それとは別の何らかのコーパス（Wikipedia など）により「automobile」と「motorcar」の間の類似度が首尾よく求められていれば，その類似度に基づいて，それぞれの文書の表現を「automobile, motorcar」「motorcar, automobile」と拡張するのは容易である（下線部分が追加された語）。その結果，2 つの文書の類似度はゼロにはならない（この場合には 1.0）。

これは，語クラスタリングの応用，もしくは情報検索におけるクエリ拡張 (query expansion) の適用と捉えることができる。その際に生じる問題は，多義的な語の扱いである。例えば，2 つの文書が「Mercury, astronomy」「Earth, space」の場合，両者に「planet（惑星）」が自動追加されれば，これらの類似度は 0.0 よりも大きくなる。しかし，「Mercury」は，神話のマーキュリーをも意味し，そのために「mythology（神話）」が加えられたとすれば，前者の文書は神話に関するものとの類似度が高くなる可能性がある（これは望ましくない）。つまり，この例では，「Mercury」の文脈語である「astronomy」を考慮して，それが示す文脈に適合した追加語を選択する工夫が必要となる。

本研究の目的は，当該文書中で各語が使用される文脈を考慮した上で，語間の類似度に基づき，短いテキストに対して語を追加する方法を考案し，それが文書クラスタリン

[1] 慶應義塾大学文学部
Faculty of Letters, Keio University, Minato-ku, Tokyo 108–8345, Japan
[2] 慶應義塾大学大学院文学研究科後期博士課程
Graduate School of Letters, Keio University
[a)] kz_kishida@keio.jp

グの性能に与える影響を実験により検証することにある（Reuters の記事を使用）。特に，本稿では，トピックモデルである潜在ディリクレ配分（LDA）および統計的翻訳のための IBM Model 1 を応用した手法を提案する。この手法では「文脈付き」の類似度が求められ，それは非対称となるため（「Mercury」が与えられたときの「planet」の類似度と，「planet」が与えられたときの「Mercury」のそれとは異なる），これを文脈に基づく非対称な単語類似度（context-based asymmetric word similarity: CAWS）と呼んでおくこととする。

以下，2 節では関連研究を概観し，3 節にて，本研究での提案手法を説明する。それに対する実験の手順と結果は 4 節で述べる。

2. 関連研究

本節では，最初に情報検索におけるクエリ拡張の方法について述べ，次に，単語間の類似度を計算する方法に関連した研究を概観する。

2.1 情報検索におけるクエリ拡張

N 件の文書から成る集合を考え，1 件の文書を d_i と表記する（$i = 1, \ldots, N$）。それらに対して標準的なテキスト処理を施し，各語の重みを要素とするベクトルに変換した結果を

$$x_i = [w_{i1}, w_{i2}, \ldots, w_{iM}]^T \tag{1}$$

と書く。ここで，M は文書集合に含まれる語の異なり総数であり，w_{ij} は，d_i における語 t_j の重みを示す。

何らかの方法によって 2 つの語 t_j と t_k の間の類似度 s_{jk}（$j, k = 1, \ldots, M$）を計算することにより得られた類似度行列 $S = [s_{jk}]$ を使えば，元の文書ベクトル x_i を

$$\tilde{x}_i = Sx_i \tag{2}$$

のように変換できる。これは，情報検索における古典的なクエリ拡張であり，この場合には，x_i は文書ベクトルではなく利用者の情報要求を表したベクトル（クエリベクトル）に相当する。例えば，クエリ中には「motorcar」が出現せず，その重みがゼロとなる場合でも（$w_{ij} = 0.0$），(2) 式によって，

$$\tilde{w}_{ij} = \sum_{k=1}^{M} s_{jk} w_{ik} \tag{3}$$

と変換されるので，「automobile」のような「motorcar」との類似度がゼロではない語がクエリに含まれていれば，「motorcar」の重み $\tilde{w}_{ij}$ は 0.0 よりも大きくなる。これは「motorcar」がクエリに新たに追加されたことに相当する。

実際のクエリ拡張では，(2) 式を直接使うのではなく，(3) 式の値が大きな語のみを追加する。実際，Zazo ら（2005）[14] は，通常のクエリ拡張の実験において，追加の候補となる語を (3) 式の値の降順に並べ，上位の語のみを加えている。この場合には，当該文書の内容と文脈的に異なる語が追加されることをある程度防ぐことができる。上記の例「Mercury, astronomy」ならば，追加される語（例えば「planet」）は文脈語「astronomy」との類似度もそれなりに高くなければならず，その結果，「神話」そのものに関連する語の追加が抑制される可能性がある。

2.2 語間の類似度の計算

クエリ拡張の実験において Zazo ら (2005)[14] が用いた類似度の計算法は，文書 d_i 中の語 t_j の重み w_{ij} を活用したものである。すなわち，(1) 式のように 1 件の文書に対して M 個の w_{ij} を並べるのではなく，1 つの語に対して N 個の w_{ij} を並べることにより，語ベクトル（word vector）を構成し，2 つの語のベクトル間での類似度を計算する。語 t_j についての語ベクトルを w_j と表記すれば，

$$w_j = [w_{1j}, w_{2j}, \ldots, w_{Nj}]^T \tag{4}$$

となり，通常，余弦係数 (cosine coefficient) によって語の間の類似度が求められる。余弦係数は，

$$s_{jk} = \frac{w_j^T w_k}{||w_j|| \cdot ||w_k||} \tag{5}$$

で定義される。なお，$||\cdot||$ はベクトルのノルムである。

なお，(4) 式中の重み w_{ij} は通常，tf-idf に基づいて計算される。例えば，最も単純な計算式は，d_i 中の語 t_j の出現回数を x_{ij}，語 t_j が含まれる文書数を n_j として

$$w_{ij} = x_{ij} \log \frac{N}{n_j} \tag{6}$$

である。(4)〜(6) 式によって求められる類似度は，コーパスからシソーラスを自動構築する試みにおいて多用されてきた（すなわち類似度シソーラス）。これを用いた初期的な研究としては Qiu と Frei(1993)[12]，最近の例としては Mohsen ら (2018)[11] が挙げられる。既出の Zazo ら（2005）[14] もまた，この方法で類似度を求めている。

一方，(4) 式以外の何らかの方法で語ベクトルを構成できれば，(5) 式によって類似度を求めることができる。例えば，Kadowaki と Kishida(2020)[6] は，Word2Vec で計算された 100 次元の語ベクトルを使って英単語間の類似度を計算し，他の方法による結果と比較している。

より一般には，(4) 式のベクトルではなく，語の共起頻度に基づいて類似度を計算する場合も多い。その際，文書中のどの範囲で共起を測定するのかが問題となり（例えば，Mandala ら (1999)[10] を参照），「1 つの文」「複数の文の中」「文書全体」などの選択肢がある。最終的な類似度は，そのようにして求められた共起頻度をそれぞれの語の出現頻度により補正して求められる。類似度の測定に相

互情報量（mutual information: MI）を活用する場合にも，共起頻度が必要となる。MI に関連する類似度については，Dagan ら (1999)[3] が包括的に論じている。

2.3 非対称な類似度

余弦係数やジャカール係数などは対称な類似度であるのに対して，語 A と B に対して，「A→B」と「B→A」とで異なる類似度を割り当てることがある。この種の非対称な（asymmetric）類似度については Kotlerman ら (2010)[9] が詳しい。例えば，非対称の類似度ならば，上位語「sports」を語 A，下位語「baseball」を語 B として，「A→B」よりも「B→A」のほうの値が大きくなると予想され，これを使っての語彙に関する推論が可能になる [9]。

非対称な類似度にはさまざまなものが存在するが，比較的単純で，なおかつ Kotlerman ら (2010)[9] による実験の中でかなり高い性能を示した計算式として，Clarke(2009)[1] が示した

$$s_{j|k} = \frac{\sum_{i:d_i \in T_j \cap T_k} \min(w_{ij}, w_{ik})}{\sum_{i:d_i \in T_k} w_{ik}} \tag{7}$$

が挙げられる。ここで，T_j と T_k はそれぞれ，語 t_j および語 t_k が出現する文書の集合を意味する。なお，非対称類似度を $s_{j|k}$ と表記した（一般に，$s_{j|k} \neq s_{k|j}$）。

Kotlerman ら (2010)[9] では，「Clarke による the degree of entailment measure」の意味で，(7) 式を「ClarkeDE」と呼んでいるが，本稿でもこの呼称を用いる。なお，重み w_{ij} を，語 t_j が文書 d_i に出現した場合に 1，そうでなければ 0 と定義すると，ClarkeDE はいわゆる重複係数に一致する。

3. 文脈を考慮して類似語を追加する方法

本節では，文脈に基づく非対称な単語類似度（CAWS）をまずは説明し，次に，それに基づいて，元の文書ベクトルに語を追加する方法について述べる。

3.1 文脈に基づく非対称な単語類似度（CAWS）

本稿では，次の 2 つの方法を組み合わせることにより，CAWS を計算することを提案する。

(1) LDA により各文書の主要なトピックを特定し，それを「文脈」として取り扱う。
(2) 翻訳確率を推計する IBM Model 1 を，文脈語を考慮するように拡張したモデルを用い，LDA により特定された「文脈」の下での語間の類似度を求める。

LDA をギブスサンプリングで実行する場合 [4]，サンプリングの各回で各文書に特定の潜在トピックが割り当てられるので，それを計数しておけば，その相対度数から $P(z_h|d_i)$ を推定できる。ここで z_h は h 番目の潜在トピックを意味し，潜在トピックの総数を L として $z_1, \ldots, z_L$ である。例えば，2000 回のサンプリングを繰り返したところ，文書 d_1 に z_1 が 500 回割り当てられたとすれば，$P(z_1|d_1) = 0.4$ となる。本研究では，この確率を使って，文書 d_i の文脈 $z_{h|i}$ を

$$z_{h|i} = \arg \max_{h'=1,\ldots,L} P(z_{h'}|d_i) \tag{8}$$

と定義する。

一方，Kishida と Ishita(2009)[8] は，並列コーパスから翻訳確率を推計するための IBM Model 1 を拡張し，例えば，$P(\text{“水星”}, \text{“planet”}|\text{“mercury”})$ のような確率を求めるための EM アルゴリズムを提案している。これは，日英での並列コーパス中の文の各アライメントにおいて，英文中の「mercury」が日本語文の「水星」と英文中の「planet」に対応付けられる確率を意味し，「planet」は「mercury→ 水星」への翻訳の際の「文脈」に相当する。本研究では，Kishida と Ishita(2009)[8] によるアルゴリズムを使って $P(t_j, z_h|t_k)$ を求め，これを CASW として用いる。

基本的な考え方は，単言語コーパス中の 1 つの文を，それぞれ，自分自身と並列していると見なして，そこから疑似的に，翻訳元と翻訳先の語のペアを生成することである。例えば，「The Mercury and another planet were observed by a telescope.」ならば，名詞に限定することとして，「mercury → planet」「mercury → telescope」「planet → mercury」「planet → telescope」「telescope → mercury」「telescope → planet」の 6 つの「翻訳」を考える（左側が「翻訳元」，右側が「翻訳先」）。単言語コーパス中の文を機械的にそのように見なせば，IBM Model 1 によって「翻訳確率」を算出でき，そしてそれは 0.0〜1.0 の値となるので，一種の類似度として解釈することが可能である。ここで，例えば，「mercury→planet」と「planet→mercury」の翻訳確率は異なるので，これによる類似度は非対称である。このように，IBM Model 1 に基づく確率を特に $P_{IBM}(\cdot)$ と表記すれば，

$$s_{j|k} = P_{IBM}(t_j|t_k) \tag{9}$$

と定義することになる。

天文学関連の潜在トピックを「T1」と書くことにすれば，Kishida と Ishita(2009)[8] の EM アルゴリズムを使って，$P(\text{“planet”}, \text{“T1”}|\text{“mercury”})$ のような文脈付きの翻訳確率を計算できる（すなわち，文脈語の代わりに，潜在トピックの記号を直接埋め込んで処理を行う）。これが本稿で提案する文脈に基づく非対称な単語類似度（CAWS）であり，(9) 式の記法に従い，

$$s_{j|k,h} = P_{IBM}(t_j, z_h|t_k) \tag{10}$$

と書く。左辺の $s_{j|k,h}$ は文脈 z_h（上の例では「T1」）において語 t_k が与えられたときの語 t_j の類似度を意味することになる。実際に，この類似度を文書 d_i のベクトル拡張に適用するには，d_i の潜在トピック $z_{h|i}$ を決めなければならない。このためには (8) 式を使う。

3.2 文脈を考慮した語の追加法

クラスタリングは「教師なし分類」であるが，本研究では，単語間の類似度を外部のデータを使って計算しておくことを想定しており，このデータを「学習用コーパス」と呼ぶ。ここではまず，学習用コーパスを使って CAWS を計算する方法を述べた後，学習用コーパスには含まれない新規の文書に，CAWS に基づいて語を追加する手順を説明する。

3.2.1 学習用コーパスでの単語間の類似度の算出

学習用コーパスを使って，文脈付き類似度を計算する手順は以下の通りである。

(1) 階層的ディリクレ過程（Hierarchical Dirichlet Process: HDP) 混合モデル（Teh ら, 2006 [13]）を学習用コーパスに適用し，潜在トピック数 L を推定する。
(2) 推定された潜在トピック数 L に基づいて，学習用コーパスに対して，LDA を実行する。
(3) LDA の結果として得られた $P(z_h|d_i)$（$h = 1, \ldots, L$; $i = 1, \ldots, N$）を計算し，(8) 式により，各文書に潜在トピックを 1 つ割り当てる。
(4) 当該文書の潜在トピックをそれに含まれるテキストに「文脈語」として組み込み，学習用コーパスを疑似的な並列コーパスと見なして，Kishida と Ishita(2009)[8] の EM アルゴリズムを実行する。

HDP 混合モデルにより $P(z_h|d_i)$ を求めることは可能であるが，本稿の実験では，HDP 混合モデルは L の推定のみに利用し，確定した L に対して，改めて，十分な反復回数の下で LDA を実行することとした。ここで，LDA のギブスサンプリングの結果を使って，$P(z_h|t_j)$ も同時に計算しておく（$h = 1, \ldots, L$; $j = 1, \ldots, M$）。これは，学習用コーパスには含まれない文書に対して，潜在トピックを決める際に利用する（後述）。

3.2.2 新規文書への語の追加

上記の手順で計算された文脈付きの類似度を使って，文書ベクトル（またはクエリベクトル）を拡張するには，学習用コーパスにはこの新規文書が含まれないため，別の方法でその潜在トピックを決定しなければならない。簡便な方法は，新規文書中に含まれるすべての語に対して，学習用コーパスから求めておいた $P(z_h|t_j)$ を合計し，その値が最も大きな潜在トピックを当該文書に割り当てることである。すなわち，新規文書 d_a 中に含まれる語の集合を T_a として，

$$z_{h|a} = \arg \max_{h'=1,\ldots,L} \sum_{k:t_k \in T_a} P(z_{h'}|t_k) \tag{11}$$

を当該文書の潜在トピックとする。なお，T_a 中の語が学習用コーパスに含まれない場合には，当然，その語は潜在トピックの決定には何ら影響しない（類似度の追加の場合も同様）。

類似語を追加する手順は，クラスタリングに LDA を使うため，従来のクエリ拡張とはやや異なっている（k-means 法などを使う場合には，従来手法を用いればよい）。まず，新規文書のベクトルでは，(6) 式ではなく，tf のみを使う。従って，拡張前の状態では，新規文書の集合を，$D = \{d_a\}$（$a = 1, \ldots, N'$）とすれば，

$$x_a = [x_{a1}, x_{a2}, \ldots, x_{aM'}]^T, \quad a = 1, \ldots, N' \tag{12}$$

である。ここで M' は，学習用コーパスと新規文書集合 D との和集合に含まれる語の異なり総数を意味するものとする。このベクトルを拡張するには，まず，類似度に関する閾値を λ とし，

$$r_{j|k} = \begin{cases} 0 & (j \neq k, s_{j|k} \leq \lambda) \\ s_{j|k} & (j \neq k, s_{j|k} > \lambda) \\ 1 & (j = k) \end{cases} \tag{13}$$

と定義する（文脈付きの場合には，右辺の $s_{j|k}$ を $s_{j|k,h}$ に置き換える。対称な類似度の場合には s_{jk}）。そして，新しい文書ベクトルにおける語 t_j の tf を

$$\tilde{x}_{aj} = \left\lceil \sum_{k=1}^{M'} x_{ak} r_{j|k} \right\rceil \tag{14}$$

とする（$j = 1, \ldots, M'$）。ここで $\lceil \cdot \rceil$ は小数点以下の切り上げを意味する。

4. 文書クラスタリングの実験

この実験では，文脈付きの非対称類似度（CAWS）に基づく文書ベクトルの拡張をクラスタリングに適用することにより，その有効性を確かめる。具体的には，それ以外の類似度でも文書ベクトルを拡張した上で，(12) 式の x_a（$a = 1 \ldots, N'$) に対するクラスタリングをそれぞれ実行し，それらの結果の比較評価を行う。

4.1 実験に用いるデータ

機械学習の研究用に作成された Reuters Corpus Volume1 (RCV1) の一部を抽出して使用する。各記事に付与された主題コードはクラスタリング結果の評価のみに用い，クラスタリング自体は (14) 式の tf に基づく LDA によって実行する（すなわち，確率 $P(z_h|d_a)$ が最大の潜在トピックを当該文書が属するクラスタと見なす）。この際，RCV1 のデータを次のように分けて活用する。

A) 学習用コーパス：1996 年 9 月の記事（60,343 件）の見出し（headline）と本文を類似度の計算に使用
B) 評価用集合（新規文書集合）：1996 年 10 月 1 日〜10 日の記事のうち，主題コードを 1 つだけ持つものを選び，その見出しのみでクラスタリングを実行

つまり，文書集合 A で語の類似度を算出し，それを使ったクラスタリングの結果を，文書集合 B で評価する。クラスタリングでは，各記事の見出しのみを対象とするので，これは短いテキストに対するクラスタリングに相当する。

4.2 実験で使用する類似度の種類

実験では，まずは文書ベクトルを拡張しない場合をベースラインとし，それに加えて，以下の類似度を使用して，それぞれ文書ベクトルの拡張を行った。

a) 対称な類似度，文脈なし：余弦係数 (5) 式
b) 非対称な類似度，文脈なし：ClarkeDE (7) 式
c) 非対称な類似度，文脈なし：IBM Model 1 (9) 式
d) 非対称な類似度, 文脈付き :IBM Model 1 の拡張 (10) 式

4.3 実験用システムの実装と結果の評価

LDA および HDP，IBM Model 1 については，基本的には，すべて Java でシステムを開発した。その際，テキスト処理の部分では，Stanford POS tagger（ver. 3.9.2）を使って，名詞，形容詞，副詞のみを抽出した。そののち，Porter のアルゴリズムで語幹抽出を行い，それらの語幹を単語と見なして，類似度の計算やクラスタリングを行った。

LDA と HDP 混合モデルの実行にはギブスサンプリングを使用し，ハイパーパラメータは，LDA では $\alpha = 0.15$，$\beta = 0.03$，HDP モデルでは $\alpha = 0.1$，$\beta = 0.01$，$\gamma = 0.5$ とした。またそれぞれの反復回数は 2200 回とし，このうちの最初の 100 回を burn-in period に設定した。

クラスタリングの結果を評価する指標としては，今回は nMI（normalized mutual information）のみを用いた。ただし，その正規化には，2 つの集合のエントロピーの最大（max）を使用した（詳細は Kishida,2014[7] を参照）。

4.4 実験結果

文書数などの基本統計，潜在トピック数の推定，実際に計算された類似度，クラスタリングの評価結果の順で，結果について説明する。

4.4.1 基本統計

学習用の 9 月分データ（学習用コーパス）および 10 月分データ（評価用集合）の基本統計を表 1 に示す。9 月分の記事件数は約 6 万件で，見出しと本文の両方を使っているため，平均文書長（記事 1 件あたりの延べ語数）は 111.5 と長いのに対して，評価用の 10 月分（約 6 千件）は見出しのみなので平均文書長は 5.0 であった。

表 1 基本統計

文書集合	文書数	異なり語数	平均文書長
9 月（学習用）	60,343	86,003	111.5
10 月（評価用）	6,262	5,459	5.0

4.4.2 潜在トピック数の推定

HDP 混合モデルにより，潜在トピック数を推定したところ，83 となった。これを学習用の 9 月分データにおけるトピック総数として，LDA を実行した。

4.4.3 算出された類似度の例

各種の類似度の計算にあたっては，学習用コーパスにおいて 10 文書よりも数多く共起している語のペアだけを対象とした。共起文書の少ない語のペアに対して類似度を算出すると，バイアスが含まれる可能性があり，また，これらのペアを落とすことによって，計算量を減らすことができる。

表 2 は，今回実際に計算された類似度の例であり，それぞれ，9 月分の学習用データから計算された「stock と bond」「stock と storag」の間の類似度の値を示している。「stock」には，「株」や「貯蔵」などの意味がある。前者は「bond（債券）」に関連し，後者は「storag（storage の語幹）」に対応するため，ここでは，事例としてこの 2 種類の組合せを選んだ。

表 2 類似度の例

(a) stock と bond:	stock → bond	stock ← bond
余弦（対称）	0.08943	
ClarkeDE	0.08022	0.12032
IBM（文脈なし）	0.00687	0.00533
IBM（文脈 1）*	0.20471	0.05838
IBM（文脈 2）*	N/A	N/A
(b) stock と storag:	stock → storag	stock ← storag
余弦（対称）	0.04184	
ClarkeDE	0.00701	0.23576
IBM（文脈なし）	0.00001	0.00144
IBM（文脈 1）*	N/A	N/A
IBM（文脈 2）*	0.20471	0.21518

注*：文脈 1 は潜在トピック No.42，文脈 2 は No.80

「stock」と「bond」では，対称な余弦，非対称な ClarkeDE とで，類似度の値にそれほど大きな差はない（それぞれ，0.08943，0.08022，0.12032）。一方，「stock」と「storag」の場合，余弦と「stock→storag」の値が小さいのに対して（それぞれ，0.04184，0.00701），「stock ← storag」では 0.23576 である。データ中での「stock」の出現頻度が多いため，そうではない「storag」との余弦係数が自動的に小さくなってしまうのに対して，ClarkeDE では「stock ← storag」の方向に対しては，それなりに大きな値となっている。これは重複係数などの非対称な類似度について，よく知られた性質である。

文脈なしの IBM 翻訳確率モデルの場合，値自体の大きさが，余弦係数や ClarkeDE と比較して，かなり小さい。それでも，「stock→storag」の 0.0001 に対して，「stock ← storag」の方向では 0.00144 であり，ClarkeDE と同様な傾向にはなっている。その下に示された「文脈 1」は潜在トピックの No.42，「文脈 2」は潜在トピック No.80 を意味し（ただし，この番号は毎回常に固定されるものではない），「stock ← bond」の値は，文脈 1 では 0.20471 と大きくなっている。同様に，文脈 2 では，「stock ← storag」「stock→storag」の

両方で，値はそれなりに大きい（0.20471 と 0.21518）。すなわち，文脈 1 は株や債券，あるいは「market（市場）」や「dollar（ドル）」に，文脈 2 は貯蔵や蓄積，あるいは「pork（豚）」や「beef（牛）」に関連しており，それぞれの文脈に応じた値（確率）が算出されていると考えられる。

4.4.4 クラスタリングの評価結果

表 2 に示されているように，それぞれの類似度で大きさの程度が異なっており，(13) 式における閾値 λ の値を統一的に決めることはできない。そこで本実験では，拡張によって新規に追加される語数がおおよそ 1〜2 語程度および 5 語程度になるように（ただし 1 文書あたりでの平均語数），試行錯誤で，類似度ごとの閾値を求めることとした。その上で，それぞれの類似度に基づいて，(13) 式と (14) 式によって文書ベクトルを拡張し，LDA によるクラスタリングを実行した。なお，その際の潜在トピック数は 72 とした。これは，評価用データの各記事に付与された主題コードの異なり総数である。

実際に追加された平均語数を表 3 に，クラスタリングの評価結果を表 4 に示す。なお，表 4 に示されている数値は，LDA を 10 回ほど繰り返してそれぞれ nMI の値を求め，それらを平均したものである。

表 3 追加された語数（平均）

類似度	閾値とその結果追加された語数			
余弦	閾値 0.8：	4.88	閾値 0.9：	1.71
ClarkeDE	閾値 0.5：	5.74	閾値 0.6：	2.67
IBM（文脈なし）	閾値 0.1：	4.13	閾値 0.2：	1.16
IBM（文脈付き）	閾値 0.1：	4.85	閾値 0.2：	1.95

表 4 のとおり，残念ながら，類似度を使った拡張はいずれも拡張をしない場合（ベースライン）よりも性能が劣るという結果となった。拡張しない場合の nMI は 0.345 であり，余弦・ClearkeDE・IBM（文脈付き）ではそれよりもわずかながら低かった。一方，IBM（文脈なし）は，閾値 0.1 の場合には 0.290，閾値 0.2 の場合には 0.293 であり，最も悪い結果となった。

4.5 考察

表 2 の事例からは，IBM モデル 1 に基づく文脈付きの非対称類似度（CAWS）はそれなりに合理的な類似度を算出するようにも思えるものの，実際には，それによる文書ベクトルの拡張は，短いテキストのクラスタリングに対して効果はなかった。今回の実験では，学習用と評価用のデータについては，両者ともに RCV1 を使っており，2 つのデータが異質であったとは考えにくい。一方，学習用コーパスの大きさ（今回は 9 月分約 6 万件の記事）が十分でなかったということは考えられる。より大きなデータを使えば，推定量としての類似度のバイアスが減少し，より良い結果が導かれた可能性はある。特に，CAWS の場合，約 6 万件を 83 のトピックに分割したため，それぞれの部分の標本サイズが十分ではなかったのかもしれない。

表 4 評価指標の値（nMI）

類似度	nMI	
拡張なし (BASE)	0.345	
余弦	閾値 0.8：0.330	閾値 0.9：0.332
ClarkeDE	閾値 0.5：0.324	閾値 0.6：0.320
IBM（文脈なし）	閾値 0.1：0.290	閾値 0.2：0.293
IBM（文脈付き）	閾値 0.1：0.328	閾値 0.2：0.322

ただし，良く知られているように，単語の類似度を計算した場合，定型のフレーズの構成語どうしの値が高くなる傾向がある。表 2 の「stock」に関しても，他の潜在トピックでは，「exchang（exchange）」（すなわち stock exchange）や，「market」（stock market）との類似度（確率）のほうがむしろ高く，「stock」と「storag（storage）」のような同義語の検出がそれに隠れてしまう。もちろん「stock」から「stock market」への拡張が，短いテキストのクラスタリングに対して肯定的な効果を持つ場合もあるかもしれないが，フレーズの構成語が追加されやすいという性質は，本稿で取り上げた類似度の限界なのかもしれない。

ということになれば，本稿の冒頭で述べた，もう 1 つの解決法である「語の分散表現を使って，文書を CBOW によるベクトルで表現する」が有力となる。実際，門脇（2020）[5] の実験は本稿とほぼ同じ条件でなされており，Word2Vec による CBOW での nMI の値を 0.437 と報告している（RCV1 の 1996 年 9 月分のデータで Word2Vec により単語の分散表現を求め，それを使って 1996 年 10 月上旬のデータに対して構成された CBOW に対して，Hartigan-Wong の k-means アルゴリズムを実行）。この 0.437 という数値は，本稿の実験のベースラインでの 0.345 を大きく上回る。この結果を表層的に見れば，「one-hot vector そのものに基づく仕組み」からは離れて，今後は「分散表現に基づく仕組み」に移行すべきということになるのかもしれない。

Word2Vec や GloVe で求められる分散表現は，本稿の用語を使えば「文脈なし」であり，さらに「文脈付き」の ELMo や BERT などを短いテキストのクラスタリングに応用することが考えられる。既に実例はあるようだが，ELMo や BERT の仕組みを教師なしの文書クラスタリングに応用する研究が今後さらに必要であろう。

5. おわりに

本稿では，短いテキストに対するクラスタリングの性能向上のために，単語間の類似度を使って文書ベクトルを拡張する試みについて報告した。具体的には，情報検索の分野でこれまで応用されてきた類似度シソーラスでの拡張（対称な余弦係数での拡張）に加え，非対称な類似度や，LDA と翻訳確率推計用の IBM Model 1 を応用した文脈付きの非対称類似度による拡張を検討した。残念ながら，

RCV1 を使った実験では，これらの類似度によるベクトル拡張の効果は観察されなかったが，この結果は，もう 1 つの性能向上の方法である分散表現の活用を示唆している。4.5 節で述べたように，今後は ELMo や BERT などを応用した文書クラスタリングの研究が重要となるだろう。

参考文献

[1] Clarke, D.: Context-theoretic semantics for natural language: An overview, *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pp. 112–119 (2009).
[2] Curiskis, S. A., Drake, B., Osborn, T. R. and Kennedy, P. J.: An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management*, vol.57, no.2, paper 102034 (2020).
[3] Dagan, I., Lee, L., and Pereira, F. C. N.: Similarity-based models of word cooccurrence probabilities. *Machine Learning*, Vol.34, No.1-3, pp.43–69 (1999).
[4] Griffiths, T. L. and Steyvers, M.: Finding scientific topics, *Proceedings of National Academic Science of the United States of America*, Vol.101(suppl1), pp. 5228—5235 (2004).
[5] 門脇夏紀: 単語の分散表現による文書クラスタリングの性能向上，2020 年度日本図書館情報学会春季研究集会，p.21–24 (2020).
[6] Kadowaki, N. and Kishida, K.: Empirical comparison of word similarity measures based on co-occurrence, context, and a vector space model, *Journal of Information Science Theory and Practice*, Vol.8, No.2, pp.6–17 (2020).
[7] Kishida, K.: Empirical comparison of external evaluation measures for document clustering by using synthetic data, *IPSJ SIG Technical Report*, Vol.2014-IFAT-113, pp.1—7 (2014).
[8] Kishida, K. and Ishita, E.: Translation disambiguation for cross-language information retrieval using context-based translation probability, *Journal of Information Science*, Vol.35, No.4 ,pp.481-495 (2009).
[9] Kotlerman, L., Dagan, I., Szpektor, I., and Zhitomirsky-Geffet, M. Directional distributional similarity for lexical inference, *Natural Language Engineering*, Vol.16, No.4, pp.359–389 (2010).
[10] Mandala, R., Tokunaga, T., and Tanaka H.: Combining multiple evidence from different types of thesaurus for query expansion, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 191—197 (1999).
[11] Mohsen, G., Al-Ayyoub, M., Hmeidi, I., and Al-Aiad, A.: On the automatic construction of an Arabic thesaurus, *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 243—247 (2018).
[12] Qiu, Y. and Frei, H. P.: Concept based query expansion, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 160—169 (1993).
[13] Teh, Y. W., Jordan, I., Beal, M. J. and Blei, D. M.: Hierarchical Dirichlet processes, *Journal of the American Statistical Association*, Vol. 101, pp. 1566–1581 (2006).
[14] Zazo, A. F., Figuerola, C. G., Berrocal, J. L. A, and Rodríguez, E.: Reformulation of queries using similarity thesauri, *Information Processing & Management*, Vol. 41, No.5, pp.1163 – 1173 (2005).