Vol.2014-NL-216 No.2
Vol.2014-SLP-101 No.2
2014/5/22

ベクトルのスパース化を用いた k 近傍法におけるハブの軽減

重藤 優太郎[1,a)]　新保 仁[1,b)]　松本 裕治[1,c)]

概要：近傍法は自然言語処理においてよく使われる手法の 1 つであり，文書分類や語義曖昧性解消，対訳抽出などの様々なタスクで用いられている．近年，機械学習分野において，近傍法におけるハブの存在が問題となっている．ハブとは多数のオブジェクトの近傍に含まれるオブジェクトのことを指しており，ハブの出現は近傍法の精度の低下を引き起こす．本稿ではハブの出現を抑制するためのベクトルのスパース化を提案し，近傍探索の典型的な一例である対訳抽出において，その効果を検証する．

1. 序論

1.1 背景

近傍法（Nearest neighbor method）は機械学習をはじめ自然言語処理や信号処理など種々の分野でよく使われる手法の 1 つである．例えば，自然言語処理では文書分類や語義曖昧性解消，対訳抽出などで用いられる．

近傍法は適切な類似度/非類似度を設定した後に，それに基づき近傍を計算する．探索問題ではクエリオブエジェクトに対しての近傍を計算し，近傍に含まれたオブジェクトを探索結果とする．分類問題では分類対象のオブジェクトに対して近傍を計算し，近傍に含まれるオブジェクトのラベルによって分類を行う．

近年，近傍法の問題の 1 つとして，ハブの出現が注目されている [1], [2], [3], [4]．ハブとは多数のオブジェクトと類似するオブジェクトのことを指す．近傍法におけるハブの出現は多数の近傍リストに同一オブジェクトが含まれることを意味し，結果として，近傍法の精度低下を引き起こす．

多数のオブジェクトと類似するオブジェクトの存在は直感的には想像しにくいが，Radovanović ら [1], [2] は種々のデータセットに対して近傍法を行い，実際にハブが出現していることを示すとともに，理論的な裏付けを与えた．同様に，Suzuki ら [4] は類似度に内積を用いた場合に，ハブが出現する理論的背景を示した．

[1] 奈良先端科学技術大学院大学
奈良県生駒市高山町生駒市高山町 8916-5
[a)] yutaro-s@is.naist.jp
[b)] shimbo@is.naist.jp
[c)] matsu@is.naist.jp

1.2 研究目的

本稿ではハブの出現を抑制するためのベクトルのスパース化を提案し，その効果を検証する．検証のために，本稿では近傍探索の典型的な一例である対訳抽出に取り組む．対訳抽出は一般的に高次元ベクトル（数百から数千次元）として単語を表現し，原言語の単語と目的言語の単語の類似度をこのベクトルを用いて計算する．その後，類似度に応じて順位付けを行い，最も順位の高いものを対訳として抽出する．事前実験において，対訳抽出にハブが出現していることを確認しており，本稿では対訳抽出に提案手法を適用することで，ハブの出現を抑制し対訳抽出の精度向上を目指す．

1.3 貢献

本稿の貢献を以下に示す．

- ハブの出現を抑制するために，ベクトルのスパース化を提案した．これまで提案されたハブの出現を抑制する手法 [3], [4] は類似度尺度に内積を用いた場合のみ効果があるものであった．提案手法は距離を用いた場合に，ハブの出現を抑制する効果がある．
- 提案したベクトルのスパース化は閾値をハブの観点から理論的に決定しており，クロスバリデーションなどによる経験的な決め方ではない．また，全ての要素に対して閾値を決めるのではなく，本稿のベクトルのスパース化は各オブジェクトの各要素ごとに閾値を決定するので自由度が高い．実験の結果，ハブの観点から決定した閾値が最も良い精度を得ていることを確認した．
- 対訳抽出においてハブが出現していることを観測した．対訳抽出におけるハブの出現は，多数の原言語の単語の対訳対として同一の目的言語の単語が選択され

Vol.2014-NL-216 No.2
Vol.2014-SLP-101 No.2
2014/5/22

ることを意味し，抽出精度の低下を引き起こしていると考えられる．対訳抽出にベクトルのスパース化を適用した結果，ハブの出現を抑制し，対訳抽出の精度が向上することを確認した．

1.4 本稿の構成

本稿の構成は次の通りである．2 章でハブに関する先行研究を述べる．その後，3 章でハブの発生を抑制するためのベクトルのスパース化を提案する．4 章で対訳抽出と実験設定について説明し，5 章で実験結果を示す．最後に，6 章で本稿をまとめる．

2. 関連研究

ハブの発生による近傍法の精度の低下は古くから報告されていたが，注目を集め始めたのは Radovanović ら [1], [2] による報告からである．彼らはハブの出現が次元の呪いの一側面であることを主張し，種々のデータセットでハブが出現していることを示した．

この研究の興味深い点として，セントロイドと類似したオブジェクトがハブになる傾向があるという発見がある．この発見に従って，Suzuki ら [3] はラプラシアンカーネルを用いることで，データ中のハブの出現を抑制できることを示した．ラプラシアンカーネルは，全てのオブジェクトとセントロイドとの内積が一定となり，結果として，ハブの出現を抑制することができる．また，Suzuki ら [4] は類似度尺度に内積を用いた場合，中心化 [5], [6], [7] にハブの発生を抑制する効果があることを示した．中心化はセントロイドを原点に移動するので，全てのオブジェクトとセントロイドとの内積が零となり，結果として，ハブの出現を抑制することができる．

一方で，近傍グラフを構築した後に，ハブの出現を抑制する研究も行われている．これらの研究は非対称な近傍グラフを対称な近傍グラフに変換することによってハブの発生を抑制するものである．Ozaki ら [8] は Mutual k-nearest neighbor によって直接近傍グラフの対称化を行った．Schnitzer ら [9] は Local Scaling と Global Scaling を用いて，距離と分布のそれぞれから近傍グラフの対称化を行った．また，Flexer ら [10] は Shared nearest neighbor にもハブの出現を抑制する効果があることを示した．

3. ハブの発生を抑制するためのベクトルのスパース化

この章では，データ中のハブの出現を抑制するためのベクトルのスパース化を提案する[*1]．本稿のスパース化は，素性ベクトルの要素を閾値よりも大きい要素のみ保存する操作を指す．言い換えると，閾値以下のベクトルの要素を零にする操作となる．

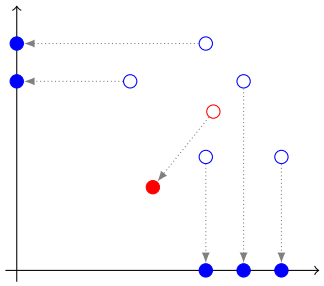

図 1 2 次元空間における，スパース化の例．（塗りつぶしていない）青丸がスパース化後のオブジェクトを示し，（塗りつぶしていない）赤丸がそのセントロイドを示す．スパース化を行うとすべてのオブジェクトは軸上の（塗りつぶした）青丸に移動し，セントロイドは（塗りつぶした）赤丸に移動する，結果的に，オブジェクトとセントロイドの距離/角度が大きくなる．

スパース化を行う事でオブジェクトを少数の軸に近づけることができ，オブジェクトの素性ベクトルが非負値のみの要素で構成される場合，各オブジェクトとセントロイドとの距離もしくは角度が大きくなる．セントロイドと類似したオブジェクトはハブになりやすい傾向にある [2] という背景から，スパース化はハブになりやすいオブジェクトを減らすことが期待できる．図 1 にスパース化の直感的な図を示す．

n 個の m-次元ベクトルが与えられた場合，スパース化の計算量は $O(mn)$ であり，また，スパース化には，メモリの容量が少なく済む上に，距離もしくは内積の計算を高速に行うことが可能となるという利点がある．

3.1 スパース化によるセントロイドとの距離の変化

上でスパース化によってハブの出現を抑制できることの，直感的な説明を行ったが，本小節ではその理論的な証明を行う．

まず，データセット $\{\mathbf{x}^{(i)} \in \mathbb{R}^m \mid i = 1, \ldots, n\}$ が与えられた場合，

$$\hat{\mathbf{x}}^{(i)} = [x_1^{(i)}\ x_2^{(i)}\ \ldots\ x_m^{(i)}]$$

セントロイド $\mathbf{c}$ は以下の式で求まる[*2]．

$$\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}$$

ここで，一般性を失うことなく $\mathbf{x}^{(1)}$ の 1 次元目をスパース化する場合を考える．スパース化したベクトルは以下のようになり，

$$\hat{\mathbf{x}}^{(1)} = [0\ x_2^{(1)}\ \ldots\ x_m^{(1)}]$$

セントロイドは

$$\hat{\mathbf{c}} = \frac{1}{n}\left\{\hat{\mathbf{x}}^{(1)} + \sum_{i\neq 1}^{n}\mathbf{x}^{(i)}\right\}$$

[*1] 以後，ベクトルのスパース化のことを単にスパース化と呼ぶ．

[*2] 上付き文字はオブジェクトの番号を表し，下付き文字はベクトルの要素の番号を表す

となる．

ここで，オブジェクト $\mathbf{x} \in \mathbb{R}^m$ とセントロイドのユークリッド距離の 2 乗は次式で与えられる．

$$\|\mathbf{x} - \mathbf{c}\|^2 = \sum_j^m (x_j - c_j)^2$$

スパース化によって，セントロイドとの距離が大きくなるためには

$$\|\hat{\mathbf{x}} - \hat{\mathbf{c}}\|^2 - \|\mathbf{x} - \mathbf{c}\|^2 \geq 0$$

が満たされればよい[*3]．この方程式を解くと，

$$0 \leq x_1^{(1)} \leq \frac{2}{n-1} \sum_{i \neq 1}^n x_1^{(i)}$$

となり，この条件を満たす要素 x_1 を 0 とすることによって，セントロイドとの距離が大きくなることがわかる．ただし，

$$\frac{2}{n-1} \sum_{i \neq 1}^n x_1^{(i)} \leq 0$$

の場合には，値域は以下のようになる[*4]．

$$\frac{2}{n-1} \sum_{i \neq 1}^n x_1^{(i)} \leq x_1^{(1)} \leq 0 \tag{1}$$

一方，全てのオブジェクトとセントロイドとの 2 乗距離の総和 D

$$D = \sum_i^n \|\mathbf{x}^{(i)} - \mathbf{c}\|^2$$

および，スパース化した後の 2 乗距離の総和 $\hat{D}$

$$\hat{D} = \|\hat{\mathbf{x}}^{(1)} - \hat{\mathbf{c}}\|^2 + \sum_{i \neq 1}^n \|\mathbf{x}^{(i)} - \hat{\mathbf{c}}\|^2$$

について考えると，スパース化した後の 2 乗距離の総和がスパース化する前の総和よりも大きくなれば，全体として，オブジェクトとセントロイドとの 2 乗距離の平均が大きくなっていることを示せる．つまり，以下のような方程式が成り立てば良い．

$$\hat{D} - D \geq 0$$

実際に，この方程式を解くと以下の解が求まる．

$$0 \leq x_1^{(1)} \leq \frac{2}{n-1} \sum_{i \neq 1}^n x_1^{(i)} \tag{2}$$

これより，式 (2) の範囲に従う，要素 $x_1^{(1)}$ を 0 にすることでオブジェクトとセントロイドの平均 2 乗距離が大きくなる．また，

$$\frac{2}{n-1} \sum_{i \neq 1}^n x_1^{(i)} \leq 0$$

の場合，値域は式 (1) となる．

[*3] 本来は距離に関して証明を行うべきだが，証明の簡単化のため，2 乗距離を用いた．
[*4] 本稿の実験（4 章）では，オブジェクトは全て第一象限に存在し，式 (2) の閾値を用いる．

3.2 スパース化のメリットとデメリット

ここまで，スパース化の説明を行ってきたが，この小節ではスパース化のメリットとデメリットを示す．まず，スパース化のメリットを以下に挙げる．

- 距離への適用: 先行研究 [3], [4] は類似度に内積を用いた場合にのみハブの出現を抑制する．これに対して，スパース化は距離を用いた場合でもハブの出現を抑制することができる．自然言語処理では類似度尺度にコサイン類似度（内積）を用いることが多いが，距離は機械学習をはじめ種々の分野で使われており，距離を用いた場合にもハブの出現を抑制できることは大きな利点だといえる．
- 閾値の決定: これまでも，高速化や省メモリといった目的でスパース化は行われていたが，ハブの出現を抑制することを目的としたスパース化は，著者らの知る限り存在しない．また，これまでのスパース化は閾値をヒューリスティックに決めていたが，本稿で提案したスパース化はセントロイドとの相対的関係に基づいて決定される．また，各オブジェクトの各次元に対して（セントロイドとの関係に）最適な閾値が決定される．そのため，全てのオブジェクトの全ての次元に対して同じ閾値を適用する必要はなく，提案手法は自由度の高い手法だといえる．

一方でデメリットとしては以下の点が挙げられる．

- 閾値の再計算: 本稿で提案したスパース化は各オブジェクトの各要素に対して式 (2) の閾値を計算する必要がある．この閾値の計算を単純に行った場合，データセット全体から再計算する必要がある．しかし，セントロイドを記憶しておけば，容易に再計算が可能であり，大きな問題とはならない．例えば，$\mathbf{x}_1^{(1)}$ をスパース化する場合，閾値は

  $$\frac{2}{n-1} \sum_{i \neq 1}^n \mathbf{x}^{(i)} = \frac{2}{n-1} \left(n\mathbf{c} - \mathbf{x}^{(1)} \right)$$

  で求まり，また，スパース化後のセントロイドの再計算も以下のように行える．

  $$\hat{\mathbf{c}} = \mathbf{c} - \frac{1}{n} \mathbf{x}^{(1)}$$

- 情報量の低下: スパース化はベクトルの要素を 0 にするので，情報量が少なくなってしまうことが考えられる．しかし，ハブの影響は情報量以上に重大な問題であり，なおかつ，提案したスパース化は基本的に，値の小さいものが 0 になるため，スパース化による情報量の低下は問題にならないと考えられる．実際に，5 章の実験結果ではスパース化することによって良い精度を得ることが確認できている．

4. 実験

本実験では提案したスパース化がハブの出現を抑制する

データセット	MEDLINE-PNE
コーパス (en)	MEDLINE
コーパス (ja)	PNE
対訳辞書	ライフサイエンス辞書
文数 (en)	139,404
文数 (ja)	512,504
対訳対の数	1,213
非対訳対の数	1,366

表 **1** データセットの統計.

かどうかを対訳抽出を通して検証する.

対訳抽出の基本的な流れは，2 言語の単語を共通のベクトル空間で表現し，その空間で近傍探索を行う．共通のベクトル空間を構築する方法は種々存在するが [11], [12], [13], [14], [15]，本実験では対訳抽出で最もよく使われるアプローチであるシード対訳対（既知の対訳対）を用いた方法 [11], [12] を採用する.

対訳抽出は共通のベクトル空間に写像した後，単純な探索問題として定式化され，原言語の単語が与えられた際に，目的言語の対訳単語を非対訳単語よりも高いランク付けを行うことが目的となる．本実験では原言語の単語に対して対訳単語が必ず一つ含まれているとする．また，2 言語のコンパラブルコーパスとシード対訳対が事前に与えられているとし，英語（原言語）から日本語（目的言語）への対訳抽出を行う.

4.1 データセット

4.1.1 コーパスと辞書

本実験では対訳抽出の評価のために英日対訳辞書とコンパラブルコーパスを用いる．対訳辞書はシード対訳対と評価用対訳対として用いる．コーパスは単言語の分布類似度の計算にのみ用いる.

- MEDLINE-PNE: 英語のコーパスとして MEDLINE [16] の 2006 年の概要の一部を用い，日本語のコーパスとして蛋白質核酸酵素（PNE） [17] の 1985 年から 2006 年まで出版された記事を用いる．対訳辞書としてライフサイエンス辞書 [18] を用いる.

表 1 にデータセットの内約を示す.

先行研究 [14], [15] に従い，各コーパスに品詞付与を行い，機能語を削除した．MEDLINE は GENIA tagger [19] を用い，PNE は MeCab [20] を用いて品詞付与を行った．日本語の品詞付与では，名詞が 2 語以上連続して付与された場合，1 語の複合名詞として取り扱った.

LSD にはコーパス中に 10 回以上出現する名詞の対訳対が 1213 対存在することが確認できた．詳しくは後述するが，この対訳対はシード対訳対と評価用対訳対に用いるため，それぞれ 2 分割する.

本実験では英語から日本語への対訳抽出を行うため，辞書に掲載されているが，日本語のコーパスにのみ 10 回以上出現した名詞を非対訳単語として用いる．対訳単語候補集合（日本語）には，これらの非対訳単語と対訳単語を含めたものを用いる．原言語（英語）は対訳単語のみを用い，システムがどれだけ目的言語（日本語）の対訳単語を高く順位付けできるかで評価を行う.

4.1.2 データ分割

本実験では，辞書をシード対訳対と評価用対訳対に分割する．辞書の全体の 60%をシード対訳対とし，残りの 40%の半分（全体の 20%）を評価用の対訳対とする．本実験ではランダムサンプリングを 4 回行い，4 回の評価の平均値を最終的な実験結果とする.

4.2 素性ベクトル

本実験では，素性ベクトルに分布類似度ベクトル [13] を採用する．ただし，Koehn ら [13] とは異なり，分布類似度の計算はコサイン類似度によって行う．分布類似度ベクトルを作るためには，文脈ベクトルを作る必要がある．先行研究 [14], [15] に従い，文脈ベクトルはコーパス中の左右 4 単語の共起単語の頻度を自己相互情報量に変換し，その正の値のみを用いた．その後，文脈ベクトルから分布類似度ベクトルを構築する．これらの操作は単言語内で行われ，分布類似度ベクトルを構築した後に，シード対訳対を用いて共通のベクトル空間に写像することで，最終的な素性ベクトルが完成する.

4.3 比較手法

本実験では以下の手法を用いる.

- *raw*: 4.2 章で説明した素性ベクトル．対訳抽出において標準的な手法であり，本実験のベースラインとなる.
- *centering*: 中心化を行った素性ベクトル．Suzuki ら [4] の分析では，（順位付けの対象となるオブジェクトではなく）順位付けの基準となるオブジェクトの分布平均（データ集合のセントロイドで近似できる）を原点に移動させることによって，ハブの出現が抑制されることを報告している．本実験では原言語の単語を基準とするので，セントロイドは原言語の評価セットから計算した.
- *sparsification*: スパース化を行った素性ベクトル．3 章で説明したとおり，スパース化の閾値は評価セットを用いて式 (2) で求める．スパース化を行う順番は貪欲的に決定した.

これらのベクトルにユークリッド距離とコサイン類似度を用いた場合のそれぞれで近傍探索を行う[*5]．ただし，中心化（*centering*）した素性ベクトルでユークリッド距離の計

[*5] 3.1 章ではユークリッド距離についてしか証明を行っていないが，対訳抽出では，類似度計算にコサイン類似度がよく使われているため，本実験ではコサイン類似度での評価も行う.

手法	ユークリッド距離		コサイン類似度	
	MRR	N_{10} skewness	MRR	N_{10} skewness
raw	15.1	7.84	15.1	7.84
centering	–	–	23.0	**3.00**
sparsification	**21.8**	**4.22**	**23.3**	3.20

表 2 対訳抽出の実験結果: Mean reciprocal rank（MRR）は値が大きいほど良く，歪度（N_{10} skewness）は値が小さいほど良い．各指標で最も良い数値を太字で表している．中心化（*centering*）を行ってもオブジェクト間の距離は変化しない（*raw* と同じ結果になる）ため，表に記載していない．

算を行っても，中心化を行わない素性ベクトル（*raw*）と同じ結果が得られるので，ユークリッド距離の結果はベースライン（*raw*）とスパース化（*sparsification*）の結果のみを示す．

4.4 評価指標

本実験では，対訳抽出を探索問題として定式化したので，評価指標として Mean Reciprocal Rank (MRR) を用いる．MRR は情報検索でよく使われる評価指標であり，以下の式によって求まる．

$$\mathrm{MRR} = \sum_{i=1}^{r} \frac{1}{\mathrm{rank}(i)} \tag{3}$$

r は評価用対訳対の数であり，rank(i) は i 番目の原単語の対訳の順位である．

本研究の目的は対訳抽出の精度とハブの関係を調べることにある．先行研究 [2], [4] に従って，ハブの影響を調べる指標として，N_{10} 分布の歪度（N_{10} skewness）を計算する．N_{10} 分布は目的言語（日本語）の単語が 10 位までに含まれた回数からなる分布であり，n が対訳候補集合の数，$x^{(i)}$ を N_{10} の値とした場合，歪度は次式で定義される．

$$(N_{10}\mathrm{skewness}) = \frac{\sum_{i=1}^{n}(x^{(i)} - \mu)^3/n}{\sigma^3} \tag{4}$$

μ と σ はそれぞれ $x^{(i)}$ の平均と分散である．N_{10} 分布の歪度が高い場合，10-近傍に同一の目的単語が複数回出現していることを意味する．N_{10} 分布は評価セットで計算しており，最終的な評価結果は 4 回のランダムサンプリングの平均値を報告する．

5. 実験結果と考察

5.1 対訳抽出の精度と歪度

表 2 に対訳抽出の精度（MRR）と 歪度（N_{10} skewness）を示す．中心化（*centering*）を行っても，オブジェクト間の距離は変化しないので（*raw* と同じ結果を得る），中心化を用いたユークリッド距離の結果は表に記載していない．

表より，ベースライン（*raw*）と比べて，スパース化（*sparsification*）の方が良い精度を得ていることが確認できる．また，中心化を用いたコサイン類似度と比べても，ほぼ同程度の精度を得ている．

a	ユークリッド距離		コサイン類似度	
	MRR	N_{10} skewness	MRR	N_{10} skewness
0.0 (raw)	15.1	7.84	15.1	7.84
0.5	15.4	7.27	15.4	7.29
1.0	18.3	5.08	17.9	5.21
1.5	21.3	**3.95**	22.4	3.75
2.0	**21.8**	4.22	**23.3**	3.20
2.5	18.4	8.48	**23.3**	2.58
3.0	12.3	12.66	22.8	2.18
3.5	8.3	14.47	19.3	2.04
4.0	5.5	15.00	17.5	1.94
4.5	3.4	14.58	15.3	1.95
5.0	2.8	14.32	13.6	1.93
5.5	2.9	14.38	12.2	1.92
6.0	2.6	14.22	10.3	**1.91**

表 3 スパース化の度合いと精度（MRR）と歪度（N_{10} skewness）: Mean reciprocal rank（MRR）は値が大きいほど良く，N_{10} skewness は値が小さいほど良い．各指標で最も良い数値を太字で表している．$a = 0.0$ の場合，スパース化を行わない（*raw*）．a が大きくなるにつれて，スパース度が大きくなっていく．$a = 2.0$ の場合が方程式による解．

歪度に注目した場合，スパース化はユークリッド距離とコサイン類似度のどちらともベースラインと比べて値が小さくなっており，スパース化によってハブの出現を抑制できていることが確認できる．中心化を用いたコサイン類似度と比較しても同等の歪度を得ていることが確認できる．

5.2 閾値と精度と歪度

最後に，スパース化の閾値について検証する．式 (2) はセントロイドとの距離を大きくするための閾値であり，精度を直接最適にするものではない．そこで，式 (2) を

$$0 \le \mathbf{x}_1^{(1)} \le \frac{a}{n-1}\sum_{i\neq 1}^{n}\mathbf{x}_1^{(i)}$$

とし，a の値を $a \in (0.0, 0.5, \ldots, 6.0)$ と変化させた場合の精度と歪度の変化を検証する．この結果を表 3 に示す．$a = 0.0$ の場合，スパース化を全く行わないので，*raw* と同じ結果を得る．

表より，$a = 2$ の場合が最も精度が良く，a を 2 よりも小さくもしくは大きくするほど MRR が低下することがわかる．

一方で，歪度は a を大きくするほどユークリッド距離では大きくなり，コサイン類似度では小さくなっている．これは，過度にスパース化を行うことによって，ほぼ零ベクトルになり，結果として，距離，類似度が意味をなさなくなるからである．

これらの結果より，式 (2) の閾値はセントロイドの関係

から決定しているが，精度の向上に繋がっていることがわかる．

6. 結論

本稿では，ハブの出現を抑制するためのベクトルのスパース化を提案した．提案したスパース化は距離を用いた場合にハブの出現を抑制することができる．また，ヒューリスティックに決めていた閾値をハブの観点から決定することが可能となる．

対訳抽出の実験の結果，スパース化がハブの発生を抑制することを確認し，その結果，精度の向上に繋がる事を確認した．また，ハブの観点から決定した閾値が最適な精度を得ていることを示した．

今後の課題には，コサイン類似度の理論的な証明を行うことや対訳抽出以外のデータに対して実験を行うことが挙げられる．

参考文献

[1] Radovanović, M., Nanopoulos, A. and Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, Vol. 11, pp. 2487–2531 (2010).

[2] Radovanović, M., Nanopoulos, A. and Ivanović, M.: On the Existence of Obstinate Results in Vector Space Models, *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10)*, pp. 186–193 (2010).

[3] Suzuki, I., Hara, K., Shimbo, M., Matsumoto, Y. and Saerens, M.: Investigating the Effectiveness of Laplacian-Based Kernels in Hub Reduction, *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12)*, pp. 1112–1118 (2012).

[4] Suzuki, I., Hara, K., Shimbo, M., Saerens, M. and Fukumizu, K.: Centering Similarity Measures to Reduce Hubs, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*, pp. 613–623 (2013).

[5] Eriksson, L., Johansson, E., Kettaneh-Wold, N., Trygg, J., Wikström, C. and Wold, S.: *Multi- and Megavariate Data Analysis Part I: Basic Principles and Applications*, Umetrics, Inc. (2006).

[6] Fisher, D. and Lenz, H.-J.: *Learning from Data: Artificial Intelligence and Statistics V*, Lecture Notes in Statistics 112, Springer (1996).

[7] Mardia, K. V., Kent, J. T. and Bibby, J. M.: *Multivariate Analysis*, Academic Press (1979).

[8] Ozaki, K., Shimbo, M., Komachi, M. and Matsumoto, Y.: Using the Mutual k-Nearest Neighbor Graphs for Semi-supervised Classification of Natural Language Data, *Proceedings of the 15th Conference on Computational Natural Language Learning (CoNLL '11)*, Portland, Oregon, USA, pp. 154–162 (2011).

[9] Schnitzer, D., Flexer, A., Schedl, M. and Widmer, G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, Vol. 13, pp. 2871–2902 (2012).

[10] Flexer, A. and Schnitzer, D.: Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?, *Data Mining Workshops, 2013 IEEE 13th International Conference on*, pp. 460–467 (2013).

[11] Rapp, R.: Automatic identification of word translations from unrelated English and German corpora, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL '99)*, pp. 519–526 (1999).

[12] Fung, P. and Yee, L. Y.: An IR approach for translating new words from nonparallel, comparable texts, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL '98)*, pp. 414–420 (1998).

[13] Koehn, P. and Knight, K.: Learning a translation lexicon from monolingual corpora, *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pp. 9–16 (2002).

[14] Tamura, A., Watanabe, T. and Sumita, E.: Bilingual lexicon extraction from comparable corpora using label propagation, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*, pp. 24–36 (2012).

[15] Vulić, I. and Moens, M.-F.: A study on bootstrapping bilingual vector spaces from non-parallel data (and nothing else), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, pp. 1613–1624 (2013).

[16] U.S. National Library of Medicine: Leasing Journal Citations (MEDLINE/PubMed including OLDMEDLINE), http://www.nlm.nih.gov/databases/journal.html (2013).

[17] PNE: PNE, http://lifesciencedb.jp/pne/ (2013).

[18] Kaneko, S., Fujita, N., Ugawa, Y., Kawamoto, T., Takeuchi, H., Takekoshi, M. and Hiroshi, O.: Life Science Dictionary: a versatile electronic database of medical and biological terms, *Dictionaries and Language Learning: How Can Dictionaries Help Human and Machine Learning*, Asian Association for Lexicography, pp. 434–439 (2003).

[19] Tsuruoka, Y.: GENIA Tagger: Part-of-speech tagging, shallow parsing, and named entity recognition for biomedical text, http://www.nactem.ac.uk/GENIA/tagger/ (2007).

[20] Kudo, T.: MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.googlecode.com/ (2013).