言語の分散表現による文脈情報を利用した言語横断情報検索

林　佑明†　酒井　哲也†

† 早稲田大学基幹理工学部情報理工学科　〒 169–8555 東京都新宿区大久保 3-4-1
E-mail: †quaternion.hysh@toki.waseda.jp, ††tetsuya@waseda.jp

あらまし　複数言語にわたる関連文書を検索する言語横断情報検索のうち，クエリ翻訳型言語横断情報検索についてはいくつかのクエリ拡張の手法が提案されている．しかし，これらの方法はクエリ翻訳の質に依存し，常に検索有効性を向上させることは難しい．そこで今回の提案手法では，そのクエリ翻訳に対するアプローチとして単語の文脈情報を強く反映すると言われる，skip-gram による分散表現を複数言語にわたって求める．そして得られた言語ベクトル間のマッピングを行い，クエリ翻訳のデータベースとして用いることで，文脈情報を含んだ言語横断情報検索の検索有効性を評価する．提案手法により，初期検索結果の質の良し悪しに対しロバストなクエリ拡張が行えることが示す．

キーワード　言語横断情報検索, 分散表現

1. はじめに

近年の情報検索技術の発達はめざましく，クエリに対してよりリッチな情報を返すエンジンが様々登場している．しかしこうした情報はクエリを基に収集されているため，クエリの言語が何かによって情報量・質共にまちまちになってしまうと考えられる．そこで，あるクエリに対してそれとは別の言語の文書を返す言語横断情報検索が考えられてきた [2]．この言語横断情報検索ではクエリを機械翻訳などを用いてターゲットの言語へ翻訳し，そのキーワードを用いてターゲット言語の文書群の検索を行う．したがってこの方法は翻訳の性能に非常に依存する．

言語横断情報検索ではより関連性の高い文書を返すために，それを構成する 2 つの要素であるクエリ翻訳とクエリ拡張において様々な改善手法が提案されてきた．

まずクエリ翻訳には大別して dictionary-based なルールベース機械翻訳及び corpus-based な統計的機械翻訳の 2 種がある．dictionary-based な方法では，着目したクエリを通常単語またはフレーズとして辞書定義に照らし合わせ，得られた定義で該当語を置換する．一方で corpus-based な方法では，2 言語の文の対応のついたパラレルコーパスの統計的情報からベイズの定理などを用いて最も確からしい翻訳結果を得る．

もう 1 つの要素であるクエリ拡張とは，何らかの方法を用いて既存の検索クエリに別のキーワードを足すことを指す．これについては pre-translation expansion(以下 pre), post-translation expansion(以下 post), 及び pre-&post- translation expansion(以下 pre&post) の 3 種の手法がある．既存の 3 種類の違いは，pre の場合クエリ翻訳前にソースの言語で行う，post の場合クエリ翻訳後にターゲットの言語で行う，pre&post では翻訳前も翻訳後もクエリ拡張を行うという点である．

ところで，上記の手法からもわかるように，これまでの機械翻訳の性能は文章によって大きく左右されてきた．というのも，文章中に異なる文脈を含意した単語があると正しい意味で訳せないからである．クエリの翻訳においてもこの「文脈」を取り込むことができれば，翻訳結果が自分の意図したものとより近くなり，返す文書の関連性も上がると考えられる．

これに関連して，word2vec [12] というツールが Google から公開された．これは skip-gram と呼ばれる言語モデルの実装で，着目語の付近の単語の出現確率を最大にするようパラメータ調整を行うことで，単語に密な分散表現を割り当てられるモデルである．ここで，分散表現とは，Hinton ら [1] によって提案された 2 つの異なる表現間の多対多の関係のことである．skip-gram モデルでは単語の分散表現を学習する．このモデルにおいて重要なのは，教師コーパスにラベルなどの正解を作る必要がないこと，そして生成された単語のベクトル空間には似た意味の単語同士は近くへ，異なる意味の単語同士は遠くへ配置されるという性質を持っていることである．すなわち，word2vec で作られた単語ベクトル 1 つ 1 つが教師コーパス中の文脈を持っているということである．

本論文では，上で述べたクエリ翻訳の中で，特に dictionary-based な方法について，これまでの手法の代替として word2vec による文脈情報を含んだ辞書を用いて言語横断情報検索の翻訳を行い，その性能を評価する．まず 2 章において関連研究を述べ，3 章において提案手法を，4 章で実験について述べる．

2. 関連研究

2.1 言語横断情報検索

情報検索の分野においては，これまでも一言語で複数言語の文書を検索しようとする試みがとられてきた．文献 [3] では言語横断情報検索において単語の翻訳時に現れる曖昧性を，クエリの拡張によって補おうとしている．この文献ではまず翻訳をする際,「クエリたちの翻訳が正しければそれらは翻訳後も共起し，逆に正しくない翻訳の場合は翻訳後の言語においてあまり共起しない」という仮説に基づいて辞書翻訳の語義曖昧性を回避しようとしている．具体的にはクエリを構成する 2 語をそれぞれ翻訳した後に対訳コーパスを利用して，ターゲット言語コーパス上の一定の窓に翻訳結果がどれだけ共起しているかをスコア化し適切な語義を割り当てるという方法をとっている．

文献 [3] ではこうした共起情報による翻訳強化とクエリ拡張

を組み合わせることで言語横断情報検索を実現している．クエリ拡張は翻訳の前に行われる pre-translation expansion 及び後に行われる post-translation expansion があり，これらは共に一度検索を行って得られた文書群からキーワードを抽出しクエリに足している．具体的には，pre の段階では上位 20 件の文書から 5 単語を選んで，それをクエリに足している．また post においては上位 30 件の文書から 50 単語を選び，それらのキーワードは通常のクエリとは別に共起度を調べられ，どの程度共起したかに基づいて検索した文書のスコアに影響を与えている．

2.2 word2vec

前章でも述べたが，word2vec は Mikolov ら [4] によって提唱された skip-gram モデルの実装である．これは着目語に対して，前後の窓に出現する単語の生起確率を最大化するようパラメータを調整する．skip-gram モデルでは，以下の式を最大化する．

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqq j\leqq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

これを図で表すと以下の様なイメージになる．

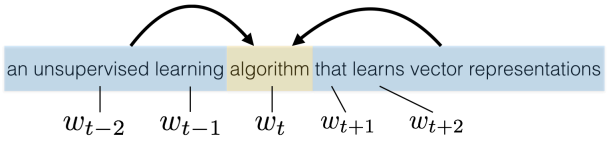

図 1　周辺語が着目語を意味づけるイメージ図

ただし，T はコーパスに含まれる単語数で，コーパスは $\{w_1, w_2, .., w_T\}$ で表される．式 (1) における p の本来の定義は

$$p(w_{t+j} \mid w_t) = \frac{\exp({v'_{w_{t+j}}}^{\mathrm{T}} v_{w_t})}{\sum_{w=1}^{W}\exp({v'_w}^{\mathrm{T}} v_{w_t})} \tag{2}$$

と表される softmax を用いるが，確率を更新する度に計算コストのかかる式の分母を算出しなければならない．そこで単語を 2 分木の葉として持ち，確率の更新をより少なく効率的に行う近似手法として提案された hierarchicasl softmax を用いる [4].

確率を定義するこの関数に用いられるパラメータベクトルを該当語に割り当てることで，語彙数を次元数に持つ sparse なベクトルではない，各要素が値を持った単語ベクトルを作ることができる．なお，word2vec では単語ベクトルの次元数はモデルの表現能力と関係しており，任意に定めることができる．しかし次元数の増加に伴って計算量も増加するためここでトレードオフが発生する．

上記モデルから作られた単語ベクトル空間においては，意味的に似た単語はベクトルとして近く，離れた意味の単語ベクトルは遠くへ配置されるという性質があることがわかっている．さらにベクトル同士の線形演算を行うと意味が遷移する例もあり，概念同士の関係も捉えていると言える．例として $Vec(\text{“}King\text{”}) - Vec(\text{“}man\text{”}) + Vec(\text{“}woman\text{”}) \quad Vec(\text{“}queen\text{”})$ が成り立つことが知られ，この例では王という概念から性別の概念を足し引きすることで女王の概念ベクトルを得ることができている．

2.3 多言語分散表現の類似性を用いた辞書作成

コーパスから得られた幾つかの言語の分散表現について，両言語で対応する意味の語のベクトルは，空間が違えどベクトル同士の位置関係を維持していることが Mikolov ら [5] によって示されている．すなわち，2 言語のベクトル空間のあいだに変換の規則性を見出すことができれば，それを用いて単語の対応がとれない (辞書に存在しない) 語に対しても相手言語のベクトル空間内でどんな値をとるかを知ることができる．Mikolov らはこのようにして，対応のわかっている少ない正解ペアから辞書になりうるより多くの単語ペアを生成することに成功している．彼らは一方のベクトル空間上のベクトルを与えることで他方のベクトル区間上のベクトルを得るために線形な写像を行い，2 乗誤差の最小化によりこれを学習している．線形な写像を用いる理由としては，ベクトルの足し引きによって遷移する意味の関係が英語もスペイン語も同様だという結果が得られたためである．すなわち回転とスケーリングによってベクトル間の対応は得られるという単純な仮定により線形写像を用いている．Neural Network などの非線形写像と比較しても，線形写像が最も良い性能を示したと報告している．

なお，Mikolov らの研究では，言語圏の近いスペイン語-英語間の他に，語族も異なり遠いと考えられる英語とベトナム語においても分散表現による翻訳を試みている．その結果，ベトナム語のモデルに多数存在する同義語の影響もあり，英語からベトナム語への 1 対 1 翻訳は精度が低いことがわかっている．

3. 提 案 手 法

従来の言語横断情報検索では，クエリの拡張においては関連単語の共起などは考慮されて来た．一方で翻訳そのものに関しては，基本的に既存辞書のエントリを参照するだけであるため一般的な定義しか得ることができない．本研究ではこれに代わり，より文脈に沿った翻訳，すなわち前章で述べた skip-gram で得られる 2 言語間分散表現の対応を用いることで，クエリを概念的に翻訳する方法を提案する．特に今回は，提案手法を英日言語横断情報検索に適用し，その効果を明らかにする。

以下に提案手法を示す。

3.1 分散表現からの辞書作成

分散表現からの辞書作成については Mikolov ら [5] に従う．具体的にはまずそれぞれ言語において単語のベクトルを得る必要があるため，前述の word2vec を用いて単語の分散表現を学習する．この際，極端な高頻度語及び低頻度語は重みを調整することで他の単語への影響を減らす.

ベクトル表現に対して次にすることは言語間のマッピングを得ることだが，この学習には訓練データが必要となる．そこで，訓練データ作成のためベクトル表現学習に用いたコーパスから頻出語を 5000 語取り出して一度 Google 翻訳 [13] による翻訳を行う．これによって得られた 5000 ペアを用いて確率的勾配降下法 [6] により変換行列のパラメータを学習する．すなわち，以下の最適化問題を解く．

$$\min_{W} \sum_{i=1}^{n} \|Wx - y\|^2 \tag{3}$$

ただしこの場合 $n = 5000$ である．上の目的関数の勾配を求め，毎ペアごとに行列 W のパラメータを更新していき，最終的に得られた W をその後の検索タスク中の pre,post で用いる．

3.2 クエリ拡張

提案手法では文献 [3] に従いクエリ拡張を行う．すなわち，一度検索した結果からキーワードを抽出する pre，post，それらを組み合せた pre&post でクエリを拡張する．pre においては本来翻訳前の言語で書かれた文書からキーワードを取り出すべきであるが，今回はデータの都合上 pre，post 共に検索対象である日本語コーパスを用いて行った場合にどうなるかを検証する．そのシステム概略を図 2 に示す．

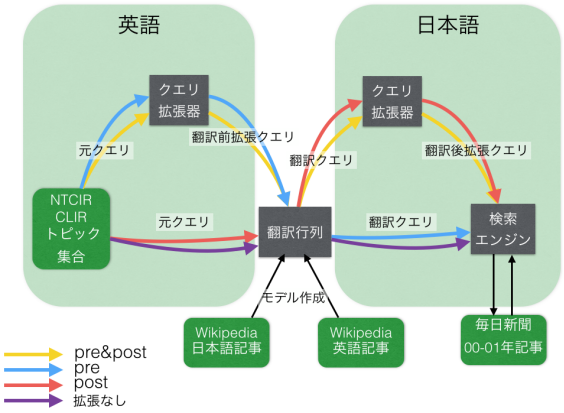

図 2　クエリ拡張の流れ

3.3 ベースラインの手法

本論文では提案手法のベースラインとして，既存の機械翻訳による手法，具体的には言語横断情報検索のクエリ翻訳の部分を Google 翻訳 [13] に置き換えたものを用いる．これにより，提案手法との差はクエリ翻訳を直接 Google 翻訳で行うか，または文脈情報を含むよう学習した対訳単語ペア群で行うか，の違いになる．すなわち Google 翻訳の場合についてもクエリ拡張を行い，提案手法との差を論じる．

4. 実　　験

4.1 デ　ー　タ

2.2 節で説明した意味の線形演算ができるような，意味的な位置関係を充分に学習した単語のベクトル表現を得るためには，巨大なコーパスが必要である．例えば文献 [5] では数百億単語を与えているが，我々の実験で学習させるデータは Wikipedia の英語・日本語記事 [15] としている．この際，Wikipedia の全記事を用いているので，語彙は分野に偏らないようになっている．単語のベクトルの次元数は文献 [5] に従ってソース言語である英語コーパスは 800 次元，ターゲット言語である日本語コーパスでは 200 次元とした．

はじめに学習用の正解として与えるデータに関しては Google 翻訳 API [13] による単語ベース翻訳を用いる．したがって英日の辞書作成時には Google 翻訳に頻出英単語を与え日本語を受け取り，日本語ベクトル空間上にその語彙があってペアを構成できる場合だけを用いている．

4.2 評価データ

評価に用いる検索対象データは NTCIR-5,6 CLIR [8] で与えられる毎日新聞 2000 年,2001 年のニュース記事を用いる．各記事が持つ情報のうち，C0 タグで表される索引記事番号を文書 ID として，T2 タグで表される本文のみを文書として扱う．評価タスクは NTCIR-5,6CLIR で用いる 97 トピックを利用する．その際，各トピックのタイトルフィールドをクエリとして扱う．

4.3 実 験 方 法

4.3.1 検索エンジン

本論文で提案している手法では，検索対象データの indexing 及びランク付き検索に Apache Solr [14] を用いている．Solr では日本語の形態素解析器として kuromoji [16] が使われている．また，ここでの検索結果のランキングは単語の tf-idf 値に基づいたスコアが計算される．その計算式を以下に示す．

$$\begin{aligned} score(q,d) =& queryNorm(q) * coord(q,d) \\ & * \sum_{t \in q} (tf(t \in d) * idf(t)^2 \\ & * t.getBoost() * norm(t,d)) \end{aligned} \tag{4}$$

上式について 1 つ 1 つ項を見ると，ある文書 d に対し $queryNorm(q)$ はクエリ同士を比較可能にするために施す正規化項，$coord(q,d)$ は文書にどれだけクエリが含まれているかのヒット率，$getBoost() * norm(t,d)$ はクエリ語 t が現れる文書の長さに対しての重み付けである．最後の項については短い文章に登場する語はより重要度が高いとみなし，より大きな重みを単語にふっている．

4.3.2 クエリ拡張方法

ベースライン，提案手法共に 3 種類のクエリ拡張で抽出するキーワードは Robertson と Sparck Jones [7] により示された offer weight によって定める．その式は全文書 N 件のうち適合とみなした文書数が R 件，キーワード候補が含まれている全文書 n 件のうち適合とみなした文書は r 件に含まれているとすると，

$$OW = r * \log\left(\frac{(r+0.5) * (N-n-R+r+0.5)}{(n-r+0.5) * (R-r+0.5))}\right) \tag{5}$$

で与えられる．今回の実験では初期検索結果の上位 $R = 10$ 件を適合文書とみなし，拡張するキーワードも 10 個とした．

4.4 評 価 指 標

検索有効性の評価指標としては nDCG を用いた．これは “normalized Discounted Cumulative Gain” の短縮語で，情報検索では広く使われる，検索結果の評価時に用いられる評価指標の 1 つである．具体的には DCG という検索順位の正しさを表す利得の和を正規化したもので，正しく文書をランキ

ングした場合 nDCG は最小値 0 から最大値 1 をとる．スコアの計算には文書の適合度に応じた利得を事前に設定する必要があり，我々の実験では高適合文書，適合文書，部分適合文書それぞれに対する利得を 3，2，1 とした．nDCG の計算には NTCIREVAL の “MSnDCG@1000” を用いた [9].

4.5 結果と考察

作ったシステムとベースラインそれぞれについて拡張なしを含む 4 種類のクエリ拡張手法で検索タスクを行なった際の nDCG を表 1 に示す.

表 1　クエリ拡張手法ごとの提案手法とベースラインの平均 nDCG の比較

	拡張なし	pre のみ	post のみ	pre&post
ベースライン	0.0931	0.0650	0.1047	0.0564
提案手法	0.0444	0.1728	0.2283	0.1733

また，ランダム化 Tukey HSD 検定 [11] を行なった結果の，提案手法でのクエリ拡張の有無に対する p 値を表 2 に示す.

表 2　提案手法同士のシステム対に対する p 値

		提案手法			
		拡張なし	pre のみ	post のみ	pre&post
提案手法	拡張なし	-	**0**	**0**	**0**
	pre のみ	-	-	0.0880	1
	post のみ	-	-	-	0.0948
	pre&post	-	-	-	-

4.5.1 クエリ拡張の平均的効果

表 1 で示した通り，クエリ拡張を行なった場合は必ず平均の nDCG の値が上昇している事がわかる．またどちらの場合も post における nDCG が最も高かった．これは pre-translation expansion を行うことによるノイズが pre や pre&post の nDCG を下げたためだと考えられる．今回の実験では翻訳前の英語クエリに対して pre を行うために最終的な検索対象である日本語の毎日新聞コーパスそのものを用いた．したがってキーワードを取り出そうとしても日本語記事中に存在するわずかな英単語を拾うことになってしまう．そのために必ずしも最もスコアの高いキーワードを選ぶことが出来ず，最適なキーワードを付加できないために検索結果が悪化したものと考えられる.

4.5.2 統計的有意性

表 2 で表されるように，提案手法のシステム同士を比べるとクエリ拡張なしとありの間には統計的有意差が得られている．しかし提案手法中の pre のみと post のみを比べると有意水準 5%では有意差が得られていない．同様のことが post のみと pre&post の間にも言え，これらについてはより多くのデータを用いた再検証が必要だと考えられる．また，pre と pre&post の間の p 値が 1 であることから，今回の実験では，pre のみのシステムにさらに post を適用しても全く効果が得られなかったと言える.

またここには載せていないが，全体の傾向としてベースラインと提案手法については有意水準 5%においておおむね有意差を得ている.

4.5.3 初期検索結果に対するクエリ拡張の有効性

表 1 において，提案手法について拡張なしに比べて拡張した場合はどれも飛躍的に検索性能が上がっている．特に post についてはおよそ 5 倍の数値をとっている．これをトピックごとにプロットしたグラフを図 3 に示す.

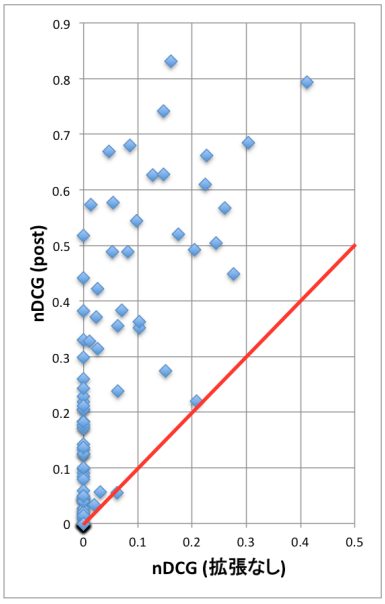

図 3　トピックごとの拡張なし時の nDCG に対する post の nDCG

この図から言えることは，まずグラフの $y \leqq x$ にあたる部分には 97 トピック中 1 トピックしかないということである．すなわち，post によってほぼ全てのクエリに対する nDCG は向上しているということである．もう 1 つが，点の分布を見てみると拡張なしにおいて nDCG が非常に悪い場合でも post によって向上しているということがわかる．すなわち提案手法と post を組み合わせることで初期検索結果の質の良し悪しに対しロバストなクエリ拡張が実現できている.

4.5.4 単言語検索とのパフォーマンス比較

言語横断情報検索システムの良さを測る指標として，単言語検索との比較が広く用いられている．そこで比較のため NTCIR CLIR トピックの日本語版を用いて，通常の日日検索及び post あり日日検索を行った．その結果の nDCG，英日検索の日日検索に対する割合を以下の表に示す.

表 3　日日検索における nDCG

	拡張なし	post
日日検索	0.5325	0.5177

表 4　英日検索の nDCG の，通常の日日検索の nDCG に対する割合

	拡張なし	pre のみ	post のみ	pre&post
ベースライン	17%	12%	20%	11%
提案手法	8%	32%	43%	33%

まず日日検索において，クエリ拡張により検索性能がわずかに悪化している．これはクエリ拡張をした影響により本来十分

な単語を含んでいたクエリにノイズとなる関連語が追加されてしまったためだと考えられる．

表 4 について，最も性能が良い時でも平均 nDCG は日日検索時の 43%と非常に低い値が出ており，提案手法では「言語横断」の部分による検索有効性の低下を防げていないと言える．ただし，トピック毎に分析を行ってみると，日日検索と比べ極端に検索有効性が低かったもの，逆に非常に検索有効性が高かったものなど，ばらつきが見られた．まず通常の日日検索に比べ特に英日検索のパフォーマンスが非常に悪かった例として，「逆エルニーニョ現象」というクエリが挙げられる．このクエリにおいてベースライン・提案手法全てのシステムで nDCG は 0 になっていたが，通常の日日検索では 1 という最大値をとっていた．このことから「逆エルニーニョ現象」のようなある種の造語，もしくは組み合わせて 1 語を形成している語の場合では翻訳の性能がすこぶる低下することがわかる．次に英日検索の post が通常の日日検索の nDCG を上回った例として,「代替エネルギー，大気汚染，電力」が挙げられる．このクエリについては，日日検索の nDCG が 0.2760 であったのに対し，post の nDCG は 0.4214 であった．理由として考えられるのは，このクエリを構成する 3 語が全て一般的な用語であることが翻訳による情報の損失を防いだということである．クエリ語が固有名詞やそれに関連する語の場合，第一にその単語の出現頻度が低いため単語の意味をうまく学習しきれない，第二に翻訳語として正確に対応する表現が得られないということが起きる．こうしたことから，1 例目のクエリにおける提案手法のパフォーマンスは悪かったが 2 例目では良かったのだと考えられる．

5. 結論と今後の課題

5.1 クエリ拡張の有効性

キーワード拡張の手法として pre と post を試みたが，post のパフォーマンスが全体的に高かった．さらに提案手法でのクエリ拡張なしの場合と比べると，初期クエリの検索結果が悪くても post によりそれをカバーして更に良いパフォーマンスを出している．このことからも提案手法は初期クエリの質に対するロバスト性があるといえる．

一方で pre の場合については，大半で post に検索性能では劣っていた．この理由としては既に述べたように，提案手法では英語キーワードを日本語記事から抽出しようとしているためである．本来，pre にはクエリと同じ言語の外部コーパスを用いるものであり，これにより今回の実験よりも検索有効性が向上できる可能性がある．これについては今後の検証が必要である．

5.2 特定ドメインに対する検索の可能性

今回は分散表現の学習に Wikipedia 全文書を用いたため，様々なドメインの記事に含まれる語に対するパラメータの調整がされている．したがってある単語との類似度の高い候補語を求めても得られる単語の分野はばらばらになると考えられる．逆にこれを特定のドメインの記事ばかり集めたコーパスを置き換えた場合，ドメインに偏った語彙に対するベクトル空間が生成されるので，そのスコープの中で類似度の高いものが候補として返されるようになる．したがって検索タスクが特定ドメインの用語を含む場合には，そのドメインの語彙に特化したモデルを用いることにより，より有効なクエリ拡張が行える可能性がある．

5.3 翻訳行列の有効性

ベースラインの Google 翻訳・クエリ拡張の組み合わせと，翻訳行列・クエリ拡張の組み合わせを比べると後者の方が性能がよく，更にこれらの間には統計的有意差も得られた．このことからクエリの語義通りの逐語訳をするのではなく，翻訳行列のような関連語彙を返すアプローチは言語横断情報検索において有効だと考えられる．

多様な語彙を与える翻訳行列は、検索意図が明確な検索タスクの場合には検索有効性の低下をもたらす可能性があるが、一方で多様化検索 [11] においてはより効果を発揮する可能性がある。これについては今後の検証が必要である．

5.4 Web 検索にむけて

文献 [5] の著者らによって 2014 年に発表された新しい文献 [10] では固定長ベクトルによりモデル化する対象が単語ではなく任意の長さのテキストに拡張されている。したがって個々の文書にこのようなベクトル表現を割り当てれば，文書同士の概念的な類似度を加味した検索が実現できる可能性がある．これを利用すれば，Web 検索時に得られる文書またはスニペット間の類似度を測ることでより多様な文書を返すことができるようになると予想される．

文　献

[1] Hinton, Geoffrey E. “Distributed representations.” 1984.

[2] Hull, David A., and Gregory Grefenstette. “Querying across languages: a dictionary-based approach to multilingual information retrieval.” *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1996.

[3] Ballesteros, Lisa, and W. Bruce Croft. “Resolving ambiguity for cross-language retrieval.” *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998.

[4] Mikolov, Tomas, et al. “Distributed representations of words and phrases and their compositionality.” Advances in Neural Information Processing Systems. 2013.

[5] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. “Exploiting similarities among languages for machine translation.” *arXiv preprint arXiv:1309.4168*. 2013.

[6] Gardner, William A. “Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique.” *Signal Processing 6.2* pp.113-133. 1984.

[7] Robertson, Stephen E., and Karen Sparck Jones. “Simple, proven approaches to text retrieval”. *UK: Computer Laboratory, University of Cambridge*. 1997.

[8] Kishida, Kazuaki, et al. “Overview of CLIR task at the fifth NTCIR workshop.” *Proc. Fifth NTCIR Workshop*. 2005.

[9] Sakai, T. “NTCIREVAL: A Generic Toolkit for Information Access Evaluation”. *FIT 2011*. Volume 2. RD-004. pp.22-30. 2011.

[10] Le, Quoc V., and Tomas Mikolov. “Distributed Representations of Sentences and Documents.” *arXiv preprint arXiv:1405.4053*. 2014.

[11] Sakai, T., “Metrics, Statistics, Tests” , PROMISE Winter School 2013:*Bridging between Information Retrieval and*

Databases (LNCS 8173). pp.116-163. Springer. 2014.
[12] word2vec, http://code.google.com/p/word2vec
[13] Google 翻訳, http://google.com/translate
[14] Apache Solr, http://lucene.apache.org/solr/
[15] Wikipedia 記事コーパス, http://dumps.wikimedia.org/
[16] kuromoji, http://www.atilika.org/