

文脈に依存した述語の同義関係獲得

柴田 知秀^{†1} 黒橋 禎夫^{†1}

本稿では「(景気が)冷え込む」と「(景気が)悪化する」のように、述語単体では同義でないが文脈に依存して同義関係となる述語ペアを自動獲得する手法を提案する。格要素と述語を組とした単位に対して、係り受け関係にある述語を要素とした素性ベクトルを構築し、分布類似度を計算することによって類似度の高いペアを同義表現として獲得する。自動生成した評価セットによる実験と人手による評価実験を行なったところ提案手法の有効性を示せた。

Context-dependent Synonymous Predicate Acquisition

TOMOHIDE SHIBATA^{†1} and SADAO KUROHASHI^{†1}

This paper proposes a method for acquiring context-dependent synonymous predicates, such as “*keiki* (business) *ga* (nom) *hiekomu* (get cold/get worse)” and “*keiki - ga - akka-suru* (deteriorate)”. The distributional similarity between combinations of a predicate and its argument is calculated, where a predicate that modifies it and that it modifies is utilized as a feature, and the pair of predicate-argument whose distributional similarity is high is adopted as a synonym. Experimental results from our evaluation set and a manual evaluation verify the effectiveness of our proposed method.

1. はじめに

自然言語処理において語の類似度計算は基本かつ重要なタスクである。語の類似度計算の方法の一つとして分布類似度というものがある。分布類似度とは「文脈の似た語は意味も似ている」という考えに基づき計算される単語の類似度である^{1),2)}。近年は Web を始めと

した大規模テキストが利用できることもあり、分布類似度を使った研究がさかんになされており³⁾、シソーラス自動構築、多義性解消などに用いられている。

分布類似度を計算する際には、多義語が問題となる。例えば「冷え込む」と「悪化する」について係り受け関係にある名詞を素性とした述語の分布類似度を計算することを考えると「冷え込む」には「寒くなる」「悪化する」などといった意味があるため「冷え込む」の素性ベクトルには「寒くなる」という意味に対する「朝晩が」「部屋が」や「悪化する」という意味に対する「景気が」「消費が」などといった素性が混在するため、結果として「冷え込む」と「悪化する」の類似度が低くなる。しかし「景気」に関する文脈中では「冷え込む」と「悪化する」の類似度は高くなるべきである。

このような多義語の問題に対して Mitchell ら⁴⁾ や Erk ら⁵⁾、Thater ら⁶⁾ はベクトル空間モデルを拡張することによって対処している。例えば、“catch a ball” という意味を表す場合に、“catch” と “ball” のベクトルを合成 (加法や乗法など) することによって文脈中の語の意味を表している。しかし、この方法では、それぞれの語のベクトルは複数の意味が混在されたまま構築されているので、文脈中の意味ではないものの影響が少なからず残ってしまうという問題がある。

本研究では、上記の研究とは異なり、ある語のある文脈中の意味を、語と文脈を組み合わせた単位に対して素性ベクトルを構築することによって直接的に表現する。本論文では、対象の多義語を述語、文脈を格要素とする。例えば、格要素「景気が」という文脈での多義語「冷え込む」の意味は「景気が冷え込む」という単位に対する素性ベクトルとする。素性としては係り受け関係にある述語を採用する。例えば「景気が冷え込む」の素性ベクトルの要素には「下落」「低迷」などがあり「景気が冷え込む」と「景気が悪化する」の素性ベクトルは同じような要素を持つことから類似度が高くなる。したがって「景気が」という文脈では「冷え込む」と「悪化する」は同義述語と認識することができる。

本論文の構成は以下の通りである。まず、2 節で関連研究について述べる。そして 3 節で素性ベクトルの構築について述べ、4 節で分布類似度計算手法について述べる。5 節では同義表現と反義関係・時間経過を区別する手法を述べる。最後に、6 節で実験結果を述べる。

2. 関連研究

2.1 分布類似度

これまでに分布類似度を計算する数多くの手法が提案されてきた。Lin は構文解析結果から分布類似度を行なう手法を提案した⁷⁾。Curran は種々の weight/measure 関数や使用す

^{†1} 京都大学
Kyoto University

るコーパスサイズなどを変化させ、実験にて有効なものを示した⁸⁾。相澤は日本語を対象として31年分の新聞記事や400万ページのWebコーパスを利用し、種々の類似度尺度やフィルタリングの効果を実験にて示した⁹⁾。Pantelらは2,000億語からなるWebコーパスを用いて5億語間の分布類似度を計算した³⁾。計算にはMapReduceを利用し、200quad-coreマシンで50時間で計算することができたと報告している。

Linらは名詞間の分布仮説を拡張し、依存構造木における二つのパス間の類似度計算を行なう手法を提案している¹⁰⁾。二つのパスが同じような名詞をつなぐ場合に類似しているとしており、例えば、“X is the author of Y”と“X wrote Y”の類似度を計算している。Szpektorらは単項のルールの類似度を計算しており、例えば、“X takes a nap”と“X sleeps”の類似度を計算している¹¹⁾。

分布類似度は自然言語処理の多くのアプリケーションまたはタスクで利用されている。Linは構文解析済みコーパスからシソーラスを構築している⁷⁾。また、LinらはCBC(Clustering By Committee)というクラスタリングを用いて単語の意味クラスを学習している¹²⁾。Chantreeらは分布類似度を用いて並列構造の曖昧性解消を行なっている¹³⁾。McCarthyらは、Linが自動構築したシソーラスを用いて語のドミナントな意味を獲得している¹⁴⁾。Brodyらは分布類似度を用いて教師なしの多義性解消手法を提案している¹⁵⁾。

2.2 ベクトル空間モデル

ベクトル空間モデルを用いて、文脈中の語の意味を捉える研究が近年行なわれている。Mitchellらは加法や乗法といったベクトル合成法を用いて句または文の意味を表す枠組みを提案している⁴⁾。Erkらは語のベクトルをsubj,objなどの関係ごとに構築する構造的なベクトル空間モデルを提案し、文脈中での語の意味を表している⁵⁾。Thaterらは、語を構文関係を利用したベクトルで表し、first-orderとsecond-orderのベクトルを組み合わせることによって文脈中の語の意味を表現している⁶⁾。

上記のモデルはいずれも語のベクトルと文脈中の語のベクトルを合成することによって文脈中での語の意味を表しているが、我々の手法では語と文脈の組み合わせに対して素性ベクトルを得ている点で異なる。

3. 素性ベクトルの構築

本節では「景気が冷え込む」といった述語項構造に対して素性ベクトルを構築する手法について述べる。まず、構文解析結果から、係り受け関係にある述語項構造ペア(PA_1, PA_2)を抽出し(PA_1 が PA_2 に係っている)、そこから、 PA_1 の素性として「 $post:PA_2$ 」、 PA_2

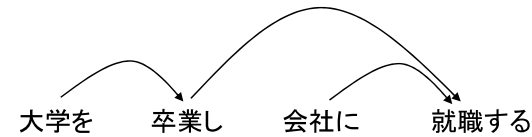


図1 構文解析結果の例

表1 素性ベクトルの例 (表中の数字はコーパスでの頻度を表す)

述語項構造	素性ベクトル
大学ヲ卒業	[素性の単位:述語] post:就職 2150, post:入社 1123, ..., pre:専攻 68, ... [素性の単位:述語項構造] post:就職 766, post:会社ニ就職 308, ..., pre:科目ヲ専攻 46, ...

の素性として「 $pre:PA_1$ 」という二つの素性を獲得する。ここで、 $post$ は素性の述語項構造の方が後ろにあることを、 pre は素性の述語項構造の方が前であることを示す。また、データスパースネスを避けるために素性の述語項構造として述語のみの場合も考える。

図1の例では述語項構造「大学ヲ卒業」の素性として、素性の単位を述語とした場合は「 $post:就職$ 」、述語項構造とした場合は「 $post:会社ニ就職$ 」が獲得され、同様に、述語項構造「会社ニ就職」の素性として、素性の単位を述語とした場合は「 $pre:卒業$ 」、述語項構造とした場合は「 $post:大学ヲ卒業$ 」が獲得される。

以上の処理をコーパス全体に対して行なうことにより述語項構造の素性ベクトルが構築される。例えば「大学ヲ卒業」という述語項構造の素性ベクトルは表1のようになる。

4. 分布類似度計算

Curranは分布類似度計算を関数 $weight$ と関数 $measure$ に分解した。関数 $weight$ は素性ベクトルの値を適切な値に変換するものであり、関数 $measure$ は関数 $weight$ で変換された値が要素であるベクトル間の類似度を計算するものである。

我々は以前、名詞の分布類似度計算において相澤の評価セット⁹⁾を用いて、以下の $weight$ 関数と $measure$ 関数の組み合わせが最も精度が高いことを示した¹⁶⁾。

$weight$ 関数:

$$weight = \begin{cases} 1(MI > 0) \\ 0(otherwise) \end{cases} \quad (1)$$

ここで、 $MI = \log \frac{P(u,f)}{P(u)P(f)}$ であり、 u は分布類似度を計算する単位、 f は素性、 $P(x)$ はコーパス中での出現確率を表わす。

measure 関数:

$$measure = \frac{1}{2}(JACCARD + SIMPSON), \quad (2)$$

ここで、

$$JACCARD = \frac{|(u_1, *) \cap (u_2, *)|}{|(u_1, *) \cup (u_2, *)|} \quad (3)$$

$$SIMPSON = \frac{|(u_1, *) \cap (u_2, *)|}{\min(|(u_1, *)|, |(u_2, *)|)} \quad (4)$$

である。

上記の weight 関数と measure 関数は名詞の分布類似度計算において最も精度が高かったものであるが、本論文で扱う述語項構造の分布類似度計算においても精度が良いと仮定し、これらの関数を用いる。

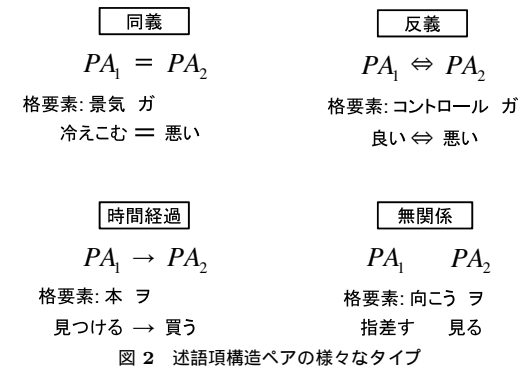
また、ここで述べた分布類似度計算は、任意の述語項構造ペアに対して適用可能なものであるが、本稿では、格要素という文脈が固定された状況での 2 述語の類似度を計算することを目的としているので、格要素は同一で述語が異なる述語項構造間の分布類似度のみを計算する。

5. 同義関係と反義・時間経過の区別

分布類似度の高い述語項構造ペアの中には同義関係のみならず、反義関係、時間経過、無関係のものも含まれる。図 2 に例をあげる。

反義関係は同義関係と同様に、同じような文脈で現れることがあり、類似度が高くなってしまふという問題が指摘されている^{17),18)}。この問題に対しては、述語が国語辞書から抽出される反義関係にあるかどうかを調べることにより、述語項構造ペアを反義関係と認識することができる。図 2 の例では、「良い」と「悪い」が国語辞典から反義関係とわかるので、「コントロール ガ 良い」と「コントロール ガ 悪い」を反義関係と認識することができる。

時間経過関係のものも同じような文脈で現れることがある。図 2 の例の「本 ヲ 見つける」と「本 ヲ 買う」の場合、共通の素性として、「pre:ブラブラする」、「post:読破」、「post:通読」などがあるために類似度が高くなってしまふ。このような時間経過関係のものは、2



つの述語項構造が係り受け関係で出現すると考えられる。この例の場合、「本を見つけて買う」という出現があるはずである。したがって、片一方の素性ベクトルの要素にもう一方の述語があるかどうかで判別することができる。この例の場合、「本を見つけて」の素性ベクトルに「post:買う」があるため、時間経過関係と認識することができる。

以上の処理によって、類似度の高い述語項構造ペアのうちから反義関係・時間経過関係を除くことができる。分布類似度の限界として、類似度は高いが同義関係ではないものが含まれてしまうという問題があり、本研究においても同義関係ではなく無関係ではあるが類似度が高い述語項構造ペアが存在する。述語項構造間に現れる節間関係の分布などの手がかりを用いて同義関係と無関係を区別するのは今後の課題とする。

6. 実験

提案手法の有効性を示すために実験を行なった。提案手法を次の二つの方法で評価した。一つ目は Mitchell らの研究⁴⁾ や Erk らの研究⁵⁾ で利用されている文類似度判定タスクであり、二つ目は人手による評価である。

まず、6.1 節でコーパスからの素性ベクトルの構築について述べる。そして、6.2 節では評価セットによる評価について、6.3 節では人手による評価について述べる。

6.1 素性ベクトルの構築

日本語約 6.5 億ページからなるコーパスを利用して実験を行なった。これは約 416 億文からなる。ウェブにはミラーページなどの重複ページが多数存在することから、約 416 億文から重複を除いた約 69 億文を実験に利用した。

表 2 評価セットと評価結果の例

述語 1	名詞	格	述語 2(正解)	類似度	述語 2(不正解)	類似度	評価
出る	家	ヲ	去る	0.283	卒業する	0	
出る	大学	ヲ	卒業する	0.393	去る	0.163	
出来る	用事	ガ	起こる	0	生まれる	0.167	×
出来る	子	ガ	生まれる	0.372	起こる	0	
飲む	敵	ヲ	圧倒する	0.116	受け入れる	0	
飲む	要求	ヲ	受け入れる	0.252	圧倒する	0	
絶やす	家系	ヲ	絶つ	0	切らす	0	
絶やす	たばこ	ヲ	切らす	0	絶つ	0	

まず、形態素解析器 JUMAN^{*1}で形態素解析を行ない、構文解析器 KNP^{*2}で構文解析を行なった。そして、構文解析結果から述語項構造の素性ベクトルを構築した。なお、頻度 1 の要素は削除した。

また、素性ベクトルを構築するコーパスサイズによる精度の変化を調べるために、69 億文のコーパスを 1/4 ずつにしたコーパスを作成し (6.9G, 1.7G, 430M, 107M, 27M), それぞれから素性ベクトルを構築した。

分布類似度計算には、4 節で述べた weight 関数と measure 関数を用いた。

6.2 評価セットによる評価

Mitchell らの研究⁴⁾ や Erk らの研究⁵⁾ で利用されている文類似度評価セットと同様の評価セットを岩波国語辞典から自動生成した。

まず、見出し語が述語であり、かつ、意味が複数存在するものを対象とする。各意味において、定義文が述語 1 語で与えられ、用例が「名詞 + 格助詞 + 述語」または「述語 + 名詞」で与えられるものを探す。例えば、「出る」の 2 番目の意味と 3 番目の意味がこの条件に該当し、辞書定義文と用例を以下に示す。

出る: 2. 去る。[用例] 家を出る。

3. 卒業する。[用例] 大学を出る。

用例が「述語 + 名詞」のものについては格が隠されているので、格解析を行なうことによって格を推定する。例えば、「重い任務」という用例の場合、述語「重い」に対する「任務」の格を推定し、「ガ」格となる。

そして、意味のペアにおいて、用例における格助詞が同一であれば、用例中の名詞と格を

*1 <http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html>

*2 <http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html>

表 3 コーパスサイズを変化させた時の精度の変化

コーパスサイズ (文数)	x			precision	recall	F
27M	4	144	0	1.000	0.027	0.053
107M	13	134	1	1.000	0.088	0.160
430M	26	120	2	0.929	0.176	0.295
1.7G	51	96	1	0.981	0.345	0.510
6.9G	77	63	8	0.906	0.520	0.661

与えられたもとの、見出しの述語に対して、自分の定義文の述語を正解とし、相手の定義文の述語を不正解とし、表 2 のような評価セットを作成する。上記の例においては、「家を」という格要素が与えられると「出る」と「去る」は同義であり、また、「大学を」という格要素が与えられると「出る」と「卒業する」は同義であるということから、表 2 の 1 行目と 2 行目の評価セットが作成される。このようなデータが 74 個 (表中の 148 行に相当) 自動生成された。

システムは「名詞-格-述語 1」と「名詞-格-述語 2(正解)」、「名詞-格-述語 1」と「名詞-格-述語 2(不正解)」の類似度をそれぞれ計算し、「名詞-格-述語 1」と「名詞-格-述語 2(正解)」の類似度の方が高くなるかどうかで性能を判定する。

正解例と誤り例を表 2 に示す。述語項構造の素性は述語単位のものとした。「」は類似度が同じ場合であり、すべてどちらの類似度も 0 であるものであった。両方とも「」であった表中の一番最後のものについては「家系 ヲ 絶やす」「家系 ヲ 絶つ」それぞれについて素性ベクトルが構築されていたが、重複する素性がなかったため、類似度が 0 になっており、「たばこヲ 切らす」も素性ベクトルが構築されていたが、「たばこ ヲ 絶やす」は素性ベクトルが構築されていなかった。

また、表 3 に素性ベクトルを作るコーパスサイズを変化させた時の精度を示す。表より、コーパスサイズを大きくするにつれて、F 値が向上し、コーパスサイズを 69 億文にした時に最もよい精度が得られたことがわかる。(以降の実験では素性ベクトルの構築に用いるコーパスサイズを 69 億文とする。)

本手法では 3 節で述べたとおり、素性として *pre* と *post* の述語を利用したが、*pre* のみ、*post* のみ利用した場合を比較した。比較結果を表 4 に示す。*pre* と *post* の両方を用いた場合が、*pre* のみ、*post* のみよりも精度が高いことがわかる。また、*pre* のみと *post* のみでは *pre* のみの方が精度が高いことがわかる。

また、述語項構造の素性として述語とした場合と述語項構造とした場合を比較した。比較結果を表 5 に示す。述語項構造を素性とした方が Precision は高いが、F 値は述語を素性と

表 4 述語項構造関係 pre と post を利用する効果

pre			post			pre + post		
precision	recall	F	precision	recall	F	precision	recall	F
0.887	0.480	0.623	0.941	0.216	0.352	0.906	0.520	0.661

表 5 素性の単位を述語項構造とした場合と述語とした場合の比較

述語項構造			述語		
precision	recall	F	precision	recall	F
0.941	0.216	0.352	0.906	0.520	0.661

表 6 人手による正解例

名詞	格	述語 1	述語 2
航空券	ヲ	購入	買う
		取る	買う
		買う	取る
	
才能	ガ	開く	開花
	
コントロール	ガ	甘い	悪い
	

した方が高いという結果が得られた。

コーパスサイズを大きくすることによってデータスパースネスの影響を軽減することができているが、依然データスパースネスの問題が残っている。この問題に対しては名詞を単語クラスに汎化して素性ベクトルを構築することによって対処できると考えられる。例えば「長雨が上がる」と「長雨が晴れる」の類似度は 0 となっているが、「長雨」と同じ単語クラスである「雨」とマージして素性ベクトルを構築することによって類似度が計算できるようになると考えられる。

6.3 人手による評価

「景気が悪化する」と「景気が冷え込む」のような同義句の場合、辞書のエントリの「冷え込む」に「悪化する」の意味が記載されていない。したがって、上記の評価セットでは評価することができない。そこで述語項構造の文脈ベクトルのうち「景気が悪化する」と「景気が冷え込む」のような格要素を共有するものに対して類似度を計算し、類似度が閾値以上のものを同義関係とみなし、人手で評価した。

まず、格要素を無作為に 20 個選び、それぞれに対して、格要素が与えられた場合に成り

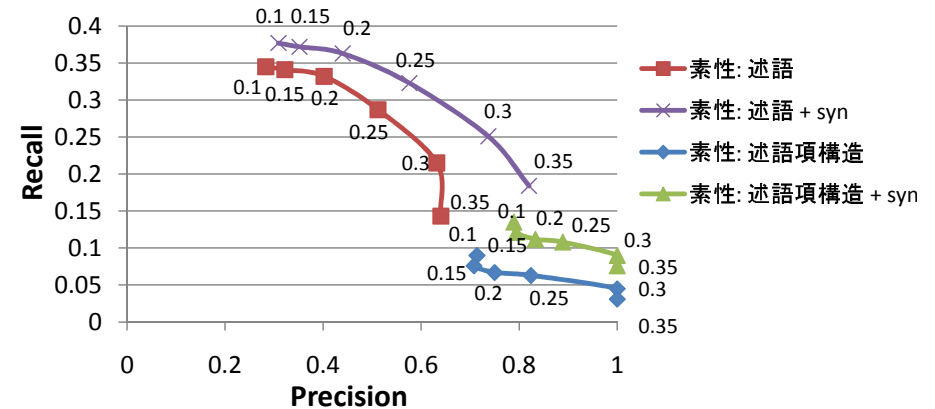


図 3 Precision-Recall 曲線 (図中の数字は同義関係とみなす閾値を表す.)

立つ同義述語ペアを人手で列挙した。例を表 6 に示す。20 個の格要素に対して合計 223 個の同義述語ペアが存在している。

システムは格要素それぞれに対して、任意の 2 述語を選び、格要素と述語の組み合わせの類似度を計算する。ただし、格要素と関連のある述語の類似度のみを計算すればよいので、「格要素 + 述語」の素性ベクトルが構築されている述語のみを対象とする。例えば、「航空券 ヲ」が格要素の場合、「航空券 ヲ 購入」、「航空券 ヲ 買う」、「航空券 ヲ キャンセル」などの素性ベクトルが構築されているので「購入」、「買う」、「キャンセル」、...などの述語から任意の 2 つの述語を選び、格要素と組にして類似度を計算する。つまり、「航空券 ヲ 購入」と「航空券 ヲ 買う」、「航空券 ヲ 購入」と「航空券 ヲ キャンセル」、「航空券 ヲ 買う」と「航空券 ヲ キャンセル」などの類似度を計算する。そして類似度が閾値以上となったものを同義関係とみなす。

図 3 に Precision-Recall カーブを示す。同義関係とみなす閾値を 0.1 から 0.35 まで 0.05 刻みで動かしたものである。素性を述語単位とした場合の閾値 0.25 の場合の F 値が最も高かった。また、「+syn」は、国語辞典から獲得された述語単体の同義関係を利用し、閾値を下回っていても同義であるとみなしたものを示す。例えば、「コントロール ガ 可能」と「コントロール ガ 出来る」の類似度は閾値以下であるが、「可能」と「出来る」が同義語であるので、これを正解とみなす。また、素性を述語項構造単位とした場合、閾値を高くすると、Recall は低い Precision は高いので、確信度の高い部分だけを同義関係として抽出するこ

表 7 反義関係除去・時間経過除去の効果

	precision	recall	F
提案手法	0.512	0.287	0.368
反義関係除去なし	0.500	0.287	0.365
時間経過除去なし	0.405	0.305	0.348

とができ、この部分を今後検索インデックスに登録し、例えばクエリが「景気が悪化する」で「景気が冷え込む」と書いてある文書とマッチするようにする予定である。

次に、5節で述べた反義関係除去、時間経過除去の効果調べた。素性の単位を述語、同義関係とみなす閾値を 0.25 とした場合で実験を行なった。反義関係除去を行わない場合は Precision のみ下がっている。時間経過除去を行わないと、Precision が下がり、Recall が上がっているが、F 値が下がっている。したがって、反義関係除去、時間経過除去ともに効果があることがわかる。

7. おわりに

本稿では「(景気が)冷え込む」と「(景気が)悪化する」のように、述語単体では同義でないが文脈に依存して同義関係となる述語ペアを自動獲得する手法を提案した。格要素と述語を組とした単位に対して、係り受け関係にある述語を要素とした素性ベクトルを構築し、分布類似度を計算することによって類似度の高いペアを同義表現として獲得した。自動生成した評価セットによる実験と人手により評価実験を行なったところ提案手法の有効性を示せた。今後の課題としては、項のクラスタリングなどを行なうことによってデータスパースネスの問題を回避することや、獲得された同義関係を検索エンジンのインデックスに利用することなどがあげられる。

参 考 文 献

- 1) J.R.Firth: *Studies in Linguistic Analysis*, chapter A synopsis of linguistic theory, Oxford (1957).
- 2) Curran, J.R. and Moens, M.: Improvements in Automatic Thesaurus Extraction, *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pp.59–66 (2002).
- 3) Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.-M. and Vyas, V.: Web-Scale Distributional Similarity and Entity Set Expansion, *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pp.938–947

- (2009).
- 4) Mitchell, J. and Lapata, M.: Vector-based Models of Semantic Composition, *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.236–244 (2008).
- 5) Erk, K. and Padó, S.: A Structured Vector Space Model for Word Meaning in Context, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp.897–906 (2008).
- 6) Thater, S., Fürstenauf, H. and Pinkal, M.: Contextualizing Semantic Representations Using Syntactically Enriched Vector Models, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp.948–957 (2010).
- 7) Lin, D.: Automatic Retrieval and Clustering of Similar Words, *Proceedings of COLING-ACL*, pp.768–774 (1998).
- 8) Curran, J.R.: From Distributional to Semantic Similarity, PhD Thesis, University of Edinburgh. College of Science (2004).
- 9) 相澤彰子: 大規模テキストコーパスを用いた語の類似度計算に関する考察, *情報処理学会論文誌*, Vol.49, No.3, pp.1426–1436 (2008).
- 10) Lin, D. and Pantel, P.: Discovery of Inference Rules for Question Answering, *Natural Language Engineering*, Vol.7, No.4, pp.343–360 (2001).
- 11) Szpektor, I. and Dagan, I.: Learning Entailment Rules for Unary Templates, *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pp.849–856 (2008).
- 12) Lin, D. and Pantel, P.: Concept Discovery from Text, *Proceedings of Conference on Computational Linguistics (COLING 2002)*, pp.577–583 (2002).
- 13) Chantree, F., Kilgarriff, A., de Roeck, A. and Wills, A.: Disambiguating Coordinations Using Word Distribution Information, *Proceedings of RANLP2005* (2005).
- 14) McCarthy, D., Koeling, R., Weeds, J. and Carroll, J.: Finding Predominant Word Senses in Untagged Text, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp.280–287 (2004).
- 15) Brody, S. and Lapata, M.: Good Neighbors Make Good Senses: Exploiting Distributional Similarity for Unsupervised WSD, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp.65–72 (2008).
- 16) 柴田知秀, 黒橋禎夫: 超大規模ウェブコーパスを用いた分布類似度計算, *言語処理学会第 15 回年次大会*, pp.705–708 (2009).
- 17) Lin, D., Zhao, S., Qin, L. and Zhou, M.: Identifying Synonyms among Distributionally Similar Words, *Proceedings of IJCAI-03*, pp.1492–1493 (2003).
- 18) Geffet, M. and Dagan, I.: The Distributional Inclusion Hypotheses and Lexical Entailment, *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp.107–114 (2005).