素性の相対性による分布類似度計算

朝倉剛史, 山本和英
長岡技術科学大学 電気系
E-mail:{asakura,yamamoto}@jnlp.org

1　はじめに

自然言語処理では単語の類似性を求めたい場面が多く存在する。単語の類似度を求めるには、シソーラスを用いる手法やコーパス内の共起語を用いる手法などが存在する。コーパス内の共起語を用いた手法として分布類似度が有名であり、多くの研究が存在する [2][3][4]。

分布類似度とは、「類似した文脈を持つ語は似ている」という『分布仮説』に基づいて計算される類似度である。つまり、それぞれの語が持つ文脈がどれだけ類似しているかという指標のもとに、語の類似度を計算する手法である。文脈から作られる素性には、あるテキスト内での共起語や同じ文中に出現する係り先の語などが挙げられる [1]。

分布類似度を計算する手法が提案されている従来研究では、各単語と関連度が低い素性を除外し関連度が高い素性のみを一貫して使用する手法がとられてきた [2][4]。しかし、単語の素性の有効性は比較対象の単語によって違うと考える。例えば、２つの単語が共通の素性を持ち、互いに関連度が低い場合はその２つの単語の性質が近いことを表しているはずである。

そこで本研究では、２単語間の情報を用いて素性の取捨選択を行う。そのために、各単語の素性に重み付けを行うだけでなく、類似度計算を行う単語対の素性を比較して相対的に素性の有効性を示す。

2　関連研究

分布類似度計算の手法として素性に着目している研究はいくつか存在する。

相澤 [1] は、各語の素性が出現頻度に影響することに注目し、２種類の低減法を提案した。１つめは、ノイズとなりうる素性を除外するために素性の出現頻度を考慮したフィルタリング法である。２つめは、単語が持つ素性の数の隔たりをなくすために単語の出現頻度を考慮したサンプリング法である。本研究ではサンプリング法は採用せず、より多くの素性について検討する。

柴田ら [2] は、超大規模コーパスを用いて分布類似度計算を行った。コーパスサイズを大きくするにつれ使用する素性の領域が広がり、精度が向上することを示した。コーパスを拡大させることで、低頻度だった単語の持つ素性が増えたことなどが要因と考えられる。本研究では低頻度の単語については今回別問題であるとして、対象外とした。

萩原ら [3] は、分布類似度計算のタスクにおける計算量の多さを問題視し、文書分類の分野における素性選択手法を適用し、素性を取捨選択した。類似度計算をする際、本当に重要な素性は限られているということを示している。本研究においても同様の視点で考察を行う。

Zhitomirsky-Geffet and Dagan[4] は、単語の素性に重み付けを行うために複数の類似した単語セットを用いた。類似度が高い単語に多く共通して出てくる素性は関連度の高い素性として重みを強くした。その結果をフィードバックして再計算を行うことで、類似度の向上を示した。本研究では、Zhitomirsky-Geffet and Dagan らの手法で重み付けを行ったのち、類似度計算を行う単語対の素性について共通している部分に着目し、重みの差を考慮する。

本手法により、類似度計算する対象の単語によって素性の有効性を示すことが可能となる。分布類似度計算における素性として、単語の共起要素である係り先を採用する。

3　分布類似度の計算

分布類似度の計算手法として、柴田らは 4 種類の Weight 関数と 5 種類の Measure 関数を用意し比較・検証した。Weight 関数は共起要素の出現頻度からノイズを低減させるフィルタリングとして用いる関数である。Measure 関数は単語間のベクトル類似度を計算する関数である。

本研究では、柴田らの検証の結果、最も精度が高かった Weight 関数と Measure 関数の組合せを用いた。最も精度が高かったのは、Weight 関数を「相互情報量 MI が閾値 β 以上の共起要素のみ使用する関数」、Measure 関数を「Simpson-Jaccard」とした時である。この柴田らの手法に加え、Zhitomirsky-Geffet and Dagan の重み付け手法による素性の重みを用いる。更にこれらの手法が着目していなかった比較対象の単語の素性にも注目することで、新たな素性選択手法を提案する。

類似度計算の流れは以下の通りである。

1. Weight 関数によるノイズ低減
2. 素性の重み付け
3. 重みを考慮した素性選択
4. Measure 関数による類似度計算

本章では柴田らの用いたそれぞれの関数についての説明、及び素性として用いるための共起要素の収集方法を示す。

3.1　Weight 関数

単語 w と共起要素 v の相互情報量を計算した。$freq(w)$ を w の頻度、$freq(w,v)$ を w と v の共起頻度とすると、相互情報量 $MI(w,v)$ の計算式は以下のように示される。

■相互情報量 MI

$$MI(w,v) = log\frac{freq(w,v)}{freq(w), freq(v)} \tag{1}$$

MI が閾値 β 以上の共起要素のみ使用し、それ以外はノイズとして除外した。閾値 β は、経験的に決めた。

3.2　Measure 関数

w_1、w_2 に対する共起要素の集合をそれぞれ V_1、V_2 とすると、Jaccard 係数及び Simpson 係数は以下のように示される。

■ Jaccard 係数

$$Jaccard(w_1,w_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \tag{2}$$

■ Simpson 係数

$$Simpson(w_1,w_2) = \frac{|V_1 \cap V_2|}{min(|V_1|,|V_2|)} \tag{3}$$

これらを用いる際の問題として、共起要素の数が大きく異なる場合に、不当に類似度が高くなるなどの弊害がある。柴田らは、この問題を解消するために、Jaccard 係数と Simpson 係数を相加平均した Simpson-Jaccard という指標を用いて、高い精度を出している。本研究もこの指標を用いて類似度 $sim(w_1,w_2)$ を求めた。

■ Simpson-Jaccard

$$sim(w_1,w_2) = \frac{Jaccard(w_1,w_2) + Simpson(w_1,w_2)}{2} \tag{4}$$

Simpson-Jaccard はあくまで共起要素集合の重なりを見ているに過ぎないため、計算時に各共起要素の頻度が考慮されない。しかし、Weight 関数で相互情報量によって、頻度が考慮されているため、本手法全体を通すと頻度が考慮されている。

3.3 共起要素の収集

柴田らの手法を習い、各単語の共起要素からテキスト内での共起語ベクトルを作成したものを素性として用いた。

まず、ある単語 w と共起関係にある単語 w' が、格要素 r でつながっている (w, r, w') の三つ組を収集した。1度しか出現しない三つ組については、ノイズとして除外した。この r と w' のペアを共起要素 v とする。r には以下のものを考える。

が, を, に, から, と, へ, まで, より, の

コーパスから収集した (w, r, w') の三つ組を用いて、すべての単語 w について共起要素 v を並べた共起ベクトルを作成した。より多くの共起要素について検討するために、一つの単語が持つ共起要素の数は制限しない。

単語 w が「しょうゆ」である場合の共起ベクトルを例 1) に示す。

例 1)「しょうゆ」の共起ベクトル

を:生産 (11), の:香り (10), の:原料 (10),
に:漬け (6), を:製造 (6), を:垂らし (5), …

「を:生産」、「の:香り」などが共起要素 v であり、括弧内の数字はコーパス内での出現頻度を示す。w には複合名詞も含む。なお、複合名詞は以下の品詞の形態素が続いた際に結合させた。

名詞-サ変接続, 名詞-一般, 名詞-固有名詞,
名詞-接尾, 接頭詞, 未知語, 記号-アルファベット

この規則により「教育制度」、「現地調査」、「ＰＴＡ」など、多くの複合名詞を同定できる。

w' の単位は1形態素とした。例えば、「しょうゆの生産手法」と「しょうゆの生産現場」という文から獲得出来る「しょうゆ」の共起要素は共に「の生産」として収集される。つまり意味的に近い共起要素は統合して抽出可能であり、単語が持つ共起要素の数の差を縮めるために有益である。

形態素解析には形態素解析器 ChaSen[(1)] を用いた。品詞体系は IPA 品詞体系辞書 [(2)] に準ずる。

4 素性の選択手法

単語に対する各共起要素の重みを計算し、求めた重みを用いて素性を取捨選択する。重みの計算手法は Zhitomirsky-Geffet and Dagan の手法を用いる。Zhitomirsky-Geffet and Dagan は重みの値が高いものだけを用いて素性選択を行っていたが、本研究では比較対象の単語と共通している共起要素について、重みの値の差を見て取捨選択することを提案する。

4.1 重み付け

ある単語 w の共起要素について、単語 w と関連度が高いものに重みを持たせたい。Zhitomirsky-Geffet and Dagan は素性の重みを判断するために、w の類似空間を作成し、同じ類似空間に含まれる単語 w_f の共起要素に注目した。

まず、シソーラスを用いて類似空間を作成する。本研究では、シソーラスとして全6階層で構成される分類語彙表 [(3)] を使用し、第4階層のレベルで同一のカテゴリに登録されているものを類義語関係にあるとする。w と類義語関係にある語 w_f を x 個収集する。

次に収集した各語 w_f について、w との類似度計算を行う。共起要素 v を持つ単語セットを $WS(v)$、w に類似する w_f の集合を $F(w)$ として、式 (5) を用いて各共起要素の重み $weight(w, v)$ を計算する。$sim(w, w_f)$ は式 (4) を用いて計算する。高い類似度を持つ w_f と多く共通していれば、重みが増すことになる。

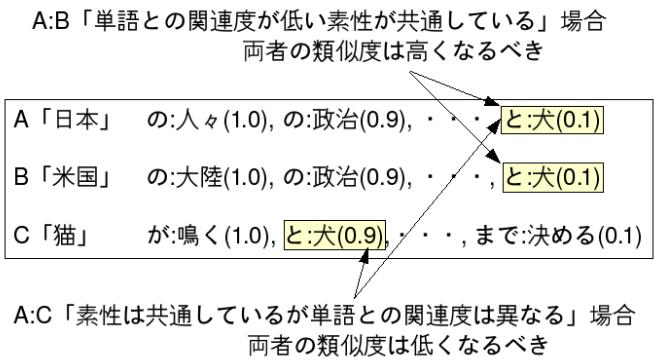

図 1: 本手法で素性選択されるべき例

■重み $weight(w, v)$

$$weight(w, v) = \sum_{w_f \in WS(v) \cap F(w)} sim(w, w_f) \qquad (5)$$

本研究では x を 5 と定め、第4階層のレベルで同一のカテゴリに登録されている単語が 5 個に満たないものは対象外とする。

4.2 素性選択

Zhitomirsky-Geffet and Dagan をはじめ多くの従来手法では、素性に重み付けを行った場合、重みの値が閾値以下のものを一律に除外していた。本手法では、類似度計算を行う際に対象となる単語の素性に注目する。共通している素性について重みの差を考慮することで、比較対象によって使用する素性を変化させ類似度を計算する。

例として、単語 A(日本)、単語 B(米国)、単語 C(猫) がそれぞれ持つ素性について重みが与えられた場合を図1に示す。ある単語 A(日本)、単語 B(米国)、単語 C(猫) が共通する共起要素 v(と:犬) を持つ場合を想定する。単語 A における v の重みが 0.1、単語 B も 0.1、単語 C は 0.9 とする。

例えば「重み 0.2 以下の素性は一律に除外する」という規則を用いた場合、単語 A と単語 B における v は除外され、残った共起要素で全ての単語との類似度を計算する。しかし、「素性が共通していて重みの差が 0.2 以上ある場合に両者を除外する」という規則に入れ換えることで、「単語 A:単語 B」の時には v が使われるが「単語 A:単語 C」の時には使われないという処理が可能となる。その結果「単語 A:単語 B」の類似度は上がり、「単語 A:単語 C」の類似度が下がることになる。特に類似度が上がるべき単語対の類似度が従来手法に比べて上がるので、精度が向上すると考える。

5 実験

5.1 実験の条件

本実験ではコーパスとして「日本経済新聞全記事データーベース 1990-2004 年度版」[(4)] を使用する。コーパスから獲得した三つ組 (w, r, w') の数は 2,584,905 件であった。1度しか出現しない三つ組は、収集時に除外したため数には含まれない。単語 w は 75,530 個獲得した。単語 w の共起要素数は一意ではない。中には共起要素の数が少ないために厳密な類似度計算が行えず、重み付けをしても有効な結果が得られないものも存在する。本研究では、低頻度ゆえに適正な類似度が求められない問題に関しては対象外とする。本研究では共起要素の数が 20 以上の単語を対象とした。

手法において類似空間を作成する必要がある。そのため単語 w は分類語彙表の第4階層の同一カテゴリ内に、多くの単語が含まれるものしか用いることができない。類語空間を作成するために、自身を含めて5個の単語が必要であると仮定し、その定義に当てはまらない単語は対象外とした。

5.2 評価セット

本研究では、相澤の評価セットに加え、相澤と同様の手法で更に難易度の高い評価セットを作成した。

相澤は、パタン法と既存の辞書を組み合わせることで、評価データを自動作成した。辞書として分類語彙表を用いている。

この評価セットには類義語ペアと非類義語ペアが存在し、各ペアの類似度から類義語であるか非類義語であるかを判定する。類似度が閾値以上の単語ペアを類義語、判定閾値 α 以下の単語ペアを非類義語と定義し、いかに精度よく分類できるかで、分布類似度計算の精度評価を行っている。判定閾値 α は経験的に決める。

評価セットは以下の手順で作成する。

1. 対象コーパスから「A や B」という定型表現のパタンを収集する
2. A:B のペアを類義語候補として抽出する
3. A と B の出現頻度が共に閾値以上のペアのみ選択する
4. A と B が共に分類語彙表の見出し語であり、かつ分類語彙表の第 4 階層のレベルで同一カテゴリに登録されているものを類義語ペアとする
5. 分類語彙表の第 2 階層で A とカテゴリが異なるもののうち、B と出現頻度が近い D を求め、A:D を非類義語ペアとする

対象コーパスから収集した定型表現を用いるのは、類語関係がコーパスなどの語彙空間に依存して決まるためである。

作成した評価セットを用いた評価方法は、類義語ペアと非類義語ペアの 2 値分類である。これはドメインが明確に分かれている分類であり、問題としては易しいと考える。理由として相澤は分類語彙表の階層的に離れた部分を使用していたからである。現に約 98 %という高い精度で 2 値分類することが可能である。

本実験では、手法の有効性を明確に示すため、難易度の高い評価セットを作成する。更に 2 段階の類義語ペアを作成するために第 3 階層、第 2 階層までのレベルで同一カテゴリに登録されているものを各自選択する。第 3 階層まで同一であるペアを「中」類義語ペア、第 2 階層まで同一であるペアを「弱」類義語ペアとする。類義語同士でも類似する程度が違うものを 2 値分類する。第 4 階層まで同一であった従来の類義語ペアを「強」類義語ペアとする。例 2) に類似度が違うそれぞれのペアの例を示す。

例 2）各類義語ペア
(「強」類義語ペア) アジア:ヨーロッパ
(「中」類義語ペア) アジア:アメリカ
(「弱」類義語ペア) アジア:我が国
(「非」類義語ペア) アジア:システム

それぞれ収集した類義語ペア集合より、4.1 に示した様に類語空間を作成するための定義に当てはまらない語を持つものは除外した。結果、「強」類義語ペア 2,310 セット、「中」類義語ペア 1,579 セット、「弱」類義語ペア 8,165 セット、「非」類義語ペア 893 セットを獲得した。

実験では、これらをランダムに 800 セットずつ抽出して評価セットとした。「強＋中」セット、「中＋弱」セット、「弱＋非」セットの 3 パターンの組合せにおける 2 値分類の F 値を求めて評価した。

5.3 Weight 関数の閾値

Weight 関数として、相互情報量が閾値 β 以下のものをノイズとして除外する。閾値 β を経験的に決めるため、閾値 β と F 値の関係について実験した。「強＋中」の評価セットにおける閾値 β と F 値の関係を図 2 に示す。

他の評価セットの組合せについても、同様の傾向がみられた。以降の実験では、各評価セットについて最も F 値が高かったときの閾値 β を用いる。各評価セットで最適だった閾値 β は「強＋中」セットでは 2.1、「中＋弱」セットでは 2.4、「弱＋非」セットでは 2.4 であった。

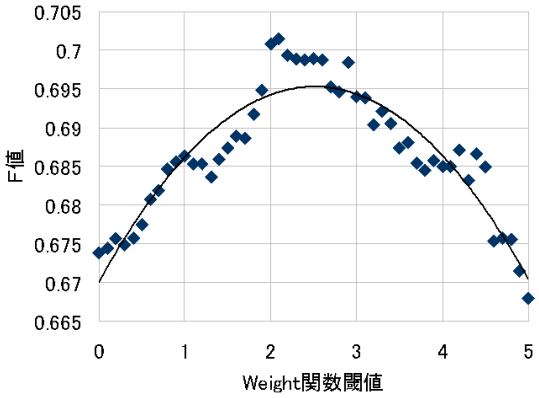

図 2: Weight 関数の閾値 β と F 値の関係

5.4 実験結果

実験は 3 手法を比較した。一つ目は柴田らの手法である。前述したように Weight 関数を用いてノイズ低減したのち、類似度計算を行った。二つ目は Zhitomirsky-Geffet and Dagan の重み付けを用いた手法である。柴田らの手法でノイズ低減を行い、残った素性に重み付けを行った。重みが閾値以下のものを除外し、更に残った素性を用いて計算を行った。三つ目は本手法である。素性の重みが閾値以下のものを全て除くのではなく、比較対象の単語によって素性を取捨選択した。 2 単語に共通する素性に注目し、各素性の重み差を考慮した。重み差が閾値以上の共起要素は共に除外した。ただし、「強＋中」セット、「中＋弱」セットについては重みがゼロのものを全て除いた上で、本手法を用いた。

以上の実験を各評価セットの組合せを用いて評価する。それぞれの手法における最大の F 値を表 1 にまとめる。

表 1: 各手法における精度 (F 値)

評価セット	柴田ら	Zhitomirsky-Geffet and Dagan	本手法
「強＋中」	0.7015	0.7908	**0.7965**
「中＋弱」	0.7474	0.7708	**0.7727**
「弱＋非」	0.8383	0.7890	**0.8395**

実験の結果「強＋中」セットのスコアは 0.7965(+0.0057)、「中＋弱」セットのスコアは 0.7727(+0.0019)、「弱＋非」セットのスコアは 0.8395(+0.0013) であった。括弧内の数字は、比較した手法のうち最も精度の高かったものからの上昇値である。柴田らの手法と Zhitomirsky-Geffet and Dagan の手法は評価セットによって優劣が異なっている。本手法ではすべての評価セットにおいて比較 2 手法と同等以上の性能を持つことを確認した。

図 3、図 4、図 5 に、重み差閾値を変化させたときの精度の推移を示す。最も精度の高かった重み差閾値は、評価セットによって大きく異なっており、「強＋中」セットでは 0.3、「中＋弱」セットでは 0.5、「弱＋非」セットでは 0.9 であった。類似度が高い領域での評価セットほど、重み差閾値が小さくなる傾向が見られる。

6 考察

評価セットを 2 値分類した結果において、類似度が高いペアを正例、低いペアを負例とする。

6.1 誤り分析

正例の誤りとして、共起要素の数が少ない低頻度語を含むペアが多く見られた。例えば「強＋中」の評価セットでは誤りのおよ

そ 2 割以上が低頻度語 (共起要素の数が 20 以下の単語) を含むペアであった。本手法では、あらかじめ用意された素性のうち、有効な素性以外を除外するため、数が少なくなってしまう。このような事例に対応するには、柴田らが提案したコーパスの超大規模化などが考えられる。また、除外するだけでなく補間するような素性選択手法があれば、この問題に対応できるだろう。

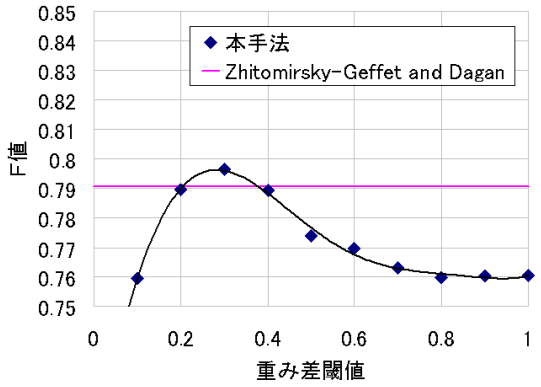

図 3: 重み差の閾値と分類精度の関係（「強＋中」セット）

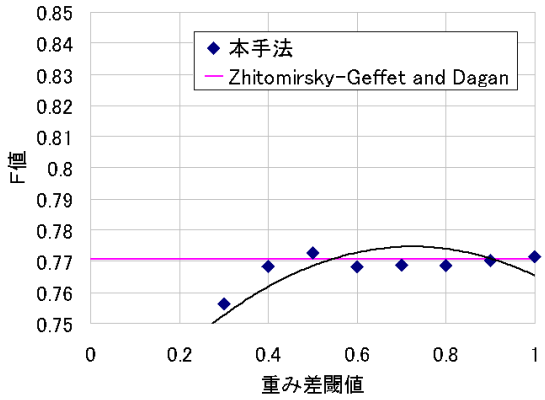

図 4: 重み差の閾値と分類精度の関係（「中＋弱」セット）

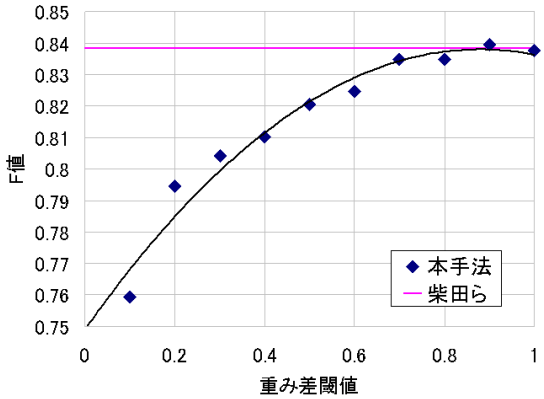

図 5: 重み差の閾値と分類精度の関係（「弱＋非」セット）

6.2 素性の削減量

本実験において、F 値が最大になる点で削減できた共起要素の割合は、「強＋中」セットで約 81 %、「中＋弱」セットで 87 %、「弱＋非」セットで 52 %であった。高い精度でありながら、多くの共起要素を削減できている。更に、どこまで精度を落とさずに、共起要素を削減できるか考察する。

萩原らは文書分類の素性選択手法で、精度の水準を著しく落とさずに素性を約 90 %削減したが、Zhitomirsky-Geffet and Dagan の重み付け手法によれば、更なる削減が可能である。

例として「強＋中」セットで比較してみる。初期状態で平均共起要素数は 608.76 個であった。閾値 0.1 刻みで変化させながら一律除外を行った際、閾値 0.8 の点で最も良い精度（F 値:0.7908）でありながら平均共起要素数を 9.53 個（約 98 %削減）まで削減可能であった。柴田らや初期状態の精度を保ちながら更なる削減も可能であった。

単語の素性のうち、単語と関連度が高く重要な素性は非常に限られていることが分かる。分布類似度計算に特化した素性選択手法ならば、それが顕著に表れる。

6.3 今後の課題

今回の実験に用いた手法では、素性の重み付けの際に類語空間を作成する必要性があることから、使用できる単語に制限がある。このような制限の無い重み付け手法を用いて有効性を示す事が求められる。また 6.1 節に示したように素性の数の問題に対応することも大きな課題である。

7 まとめ

分布類似度計算における素性の有効性は、比較対象の単語によって決まると考えた。そこで、素性の重みの値をそのまま用いるのではなく、比較対象の単語の素性にも注目した。2 単語に共通している素性について、素性の重みの差を考慮して相対的に素性を選択する手法を提案した。実験の結果、比較した 2 手法と同等以上の精度であり、手法の有効性を示した。

使用した言語資源及びツール

(1) 形態素解析器 Chasen, Ver.2.3.3, 奈良先端科学技術大学院大学 松本研究室, http://chasen-legacy.sourceforge.jp/

(2) IPA 品詞体系辞書 IPADIC,Ver.2.7.0, 奈良先端科学技術大学院大学 松本研究室, http://sourceforge.jp/projects/ipadic/

(3) 分類語彙表増補版, 国立国語研究所

(4) 日本経済新聞全記事データーベース 1990-2004 年度版, 日本経済新聞社

参考文献

[1] 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.

[2] 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. 言語処理学会年次大会, D4-7, pp.705-708, 2009.

[3] 萩原正人, 小川泰弘, 外山勝彦. 分布類似度のための文脈素性選択. 言語処理学会 NLP 若手の会第 2 回シンポジウム, 発表 11, 2007. http://yans.anlp.jp/symposium/2007/paper/hagiwara.pdf

[4] Maayan Zhitomirsky-Geffet, Ido Dagan. Bootstrapping Distributional Feature Vector Quality. Computational Linguistics, Volume 35, Issue 3, pp.435-461, 2009.