超大規模ウェブコーパスを用いた分布類似度計算

柴田 知秀　　黒橋 禎夫
京都大学大学院情報学研究科
{shibata, kuro}@i.kyoto-u.ac.jp

1　はじめに

分布類似度とは、「似た語は似た文脈で出現する」という分布仮説 [2] に基づいて計算される語の類似度であり、これまで多くの研究で分布類似度を計算する手法が提案されてきた [4]。単語の類似度を測る方法として、人手で構築・整備されたシソーラスを利用する方法が考えられるが、人手で構築・整備されたリソースは低カバレージであることや一貫性を保つことが難しいといった問題があるので、コーパスから計算された分布類似度は非常に重要である。

近年、大量の Web コーパスが使えるようになってきており、NLP のいろいろなタスクにおいて、大規模コーパスを使うことにより精度が向上している。本論文では分布類似度計算においてもコーパスサイズを大きくすることにより精度が向上するかどうかを示す。これまでの研究では例えば、Lin は 6400 万語のテキスト [4]、Curran は 20 億語のテキスト [1]、相澤は 4000 万日本語ウェブページ [6] を利用している。これらに対して、我々は 1 億ページから得られた 250 億語からなるテキストを利用する。

2　共起関係の抽出

ある単語 w が関係 r で他の単語 w' と共起していることを (w, r, w') の 3 つ組で表す。本研究では、2 語 w と w' が係り受け関係にある時を共起関係にあるとみなす。r は格要素にあたり、以下のものを考える。

ガ, ヲ, ニ, カラ, ト, ヘ, マデ, ヨリ, ノ

また、語 w' と r をペアにしたものを語 w の共起要素と呼ぶ。

係り受け解析済みコーパスから共起関係を抽出し、その結果を集計することにより、すべての名詞を、共起要素を並べた共起ベクトルで表す。図 1 に「医者」の共起ベクトルを示す。

2.1　曖昧性のある係り受けの除外

解析誤りである係り受け関係から共起関係を抽出すると分布類似度計算においてノイズとなってしまう。

ニ:行く　ニ:かかる･･･ノ:指示　ガ:青くなる　･･･
(20,399,　9,284, ･･･ 945,　944,　･･･)
図 1: 「医者」の共起ベクトル

河原らは大規模コーパスから格フレームを自動構築する際に、ヒューリスティックなルールに基づき、係り受け先に曖昧性のある係り受け関係を捨て、信頼できる係り受け関係のみを用いており [3][1]、我々も同じ手法を利用する。

2.2　語の単位

語 w として、単名詞と複合名詞を考える。例えば以下の文では単名詞として「電話」、複合名詞として「携帯電話」を抽出する。

(1)　携帯電話 を購入した。

3　分布類似度計算

Curran は分布類似度計算を weight 関数と measure 関数に分解した [1]。weight 関数は頻度を別の値に変換する関数であり、measure 関数は 2 つのベクトル間の類似度を計算する関数である。

3.1　Weight

本研究で用いた weight 関数を表 1 に示す。このうち、相互情報量 (MI) が広く用いられている。相互情報量を用いる際には低頻度語の値が高くなるという問題があるので、3 番目の関数は補正をかけたものである [5]。4 番目の関数は MI が閾値　(≥ 0) よりも大きい場合に 1、そうでない場合に 0 とする関数である。

3.2　Measure

本研究で用いた measure 関数を表 2 に示す。LIN98 は Lin によって提案された measure 関数である [4]。JACCARD、SIMPSON、SIMSON-JACCARD は weight 関数 P_β の場合に利用する。JACCARD や SIMPSON を用いる場合、共起要素の数がかなり異な

[1]河原らによるとすべての係り受け関係の精度は 90.9%、信頼できる係り受け関係の精度は 97.2%である。

表 1: Weight 関数

FREQ	$f(w,c)$
MI	$\log \frac{P(w,c)}{P(w)P(c)}$
MI′	$\frac{f(w,c)}{f(w,c)+1} \cdot \frac{min(f(w,*),f(*,c))}{min(f(w,*),f(*,c))+1} \cdot \frac{P(w,c)}{P(w)P(c)}$
P_β	1 if MI $> \beta$: otherwise 0

表 2: Measure 関数 ($(w,*) \equiv \{(c)|\exists(w,c)\}$)

COSINE	$\frac{\sum wgt(w_1,*)*wgt(w_2,*)}{\sqrt{\sum wgt(w_1,*)^2 * \sum wgt(w_2,*)^2}}$
LIN98	$\frac{\sum_{(w_1,*)\cap(w_2,*)} wgt(w_1,*)+wgt(w_2,*)}{\sum_{(w_1,*)} wgt(w_1,*)+\sum_{(w_2,*)} wgt(w_2,*)}$
JACCARD	$\frac{(w_1,*)\cap(w_2,*)}{(w_1,*)\cup(w_2,*)}$
SIMPSON	$\frac{(w_1,*)\cap(w_2,*)}{min((w_1,*),(w_2,*))}$
SIMPSON-JACCARD	$\frac{1}{2}$(JACCARD + SIMPSON)

る場合に不当に類似度が高くなるという問題が生じる。そこで本研究では JACCARD と SIMPSON を平均した SIMSON-JACCARD を利用することによって共起要素の数が違う場合の影響を軽減する。

4 実験

4.1 分布類似度計算

検索エンジン TSUBAKI[2]で検索対象となっている日本語 1 億ページをコーパスとして利用した。このコーパスは 60 億文 (1 兆語) 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、60 億文を uniq した 16 億文 (250 億語) を実験に利用した。文あたりの平均文字数、形態素数はそれぞれ 28.3、15.6 であった。

まず、ウェブコーパスを形態素解析器 JUMAN、構文解析器 KNP で解析した。この処理は 150CPU を用いて約 1 週間かかった。次に、共起関係を抽出し、名詞を共起ベクトルで表した。ここで、一回しか出現しない名詞・共起要素ペアは足切りした。

共起ベクトルは表 3 に示す 3 セットを構築し、それぞれ 5 サイズのコーパスから作成した (6.3M, 25M, 100M, 400M, 1.6G 文)。コーパスサイズを変化させた時の名詞数 (相互情報量が正のもののみ) と平均共起要素数を表 3 に示す。コーパスサイズを 4 倍ずつ増やすにつれて、名詞数が多くなるが、その倍率はだんだん小さくなっている。また、語の単位を複合名詞とすることによって、名詞数は多くなり、それに伴ない、平均共起要素数は小さくなる。

4.2 評価セット

相澤の評価セット [6] を用いて分布類似度の精度を評価した。この評価セットでは 2 語が与えられ、それらが類義語であるか非類義語であるかを判定する。評価セットは 2 つのタスクからなりそれぞれ以下の手順で作成されている (図 2 に例を示す)。

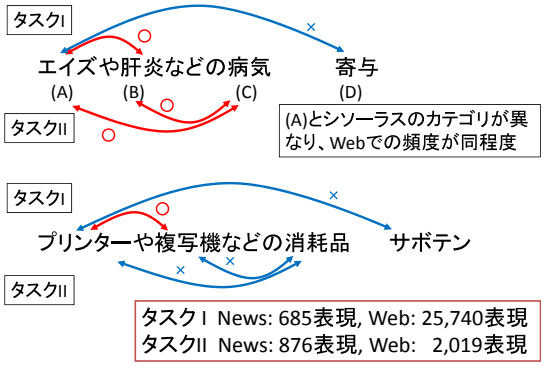

図 2: 評価セット [6] の作成方法

- タスク I
  - 「A や B などの C」というパターンに着目し、{(A), (B)} を類義語として抽出
  - 分類語彙表で (A) と異なるカテゴリに属する語のうち、Web での頻度が (B) と同程度の語 (D) を求め、{(A), (D)} を非類義語とする
- タスク II
  - 「A や B などの C」というパターンに着目し、(C) が (A) や (B) の上位概念になっているかどうかを人手で判定

(A) と (B) はタスク I では同位語、タスク II では上位下位関係になっている。二つのタスクともに、新聞と Web の二つのコーパスから作成されており、データ数を図 2 の下部に示す。また、このデータには、単名詞だけでなく複合名詞も含まれており、データの 10.4%が複合名詞である。評価セットの複合名詞の分布類似度計算を共起ベクトル (i) または (ii) で行なう際には単名詞に汎化する。例えば、「非鉄金属」という語は「金属」に汎化して分布類似度計算を行なう。

4.3 評価

予備実験の結果、weight 関数と measure 関数のあらゆる組み合わせのうち、表 4 にあげた 5 つの類似度尺度の精度が比較的よかった。これら 5 つの類似度について、F 値を 4 つの評価セットを用いて評価した[3]。この時、共起ベクトルは (ii) を用いた。F 値の評価は、類似度閾値 (語の類似度がこの値以上なら類義語とみなす) を 0.01 から 0.40 まで 0.01 ずつ変化させた時の最大値を求めて行なった。表 4 に結果を示す。P-SJ がすべての評価セットにおいて最も精度がよかったこ

[2]http://tsubaki.ixnlp.nii.ac.jp/index.cgi

[3]P_β の β の値は予備実験の結果から 2 に設定した。

表 3: コーパスサイズと名詞数・平均共起要素数の関係

	(i)		(ii)		(iii)	
語の単位 曖昧性のある係り受け	単名詞 あり		単名詞 なし		複合名詞 なし	
コーパスサイズ (文)	名詞数	平均共起要素数	名詞数	平均共起要素数	名詞数	平均共起要素数
6.3M	69,356	9.67	39,325	7.35	57,774	4.57
25M	181,218	15.80	101,341	12.83	203,379	6.14
100M	456,858	21.04	247,292	18.03	639,702	7.18
400M	1,195,702	25.29	630,725	22.47	2,100,541	7.62
1.6G	3,197,004	28.08	1,698,155	25.07	7,311,191	7.38

表 4: 5 つの類似度尺度の F 値 (括弧内の数字は最大の F 値の時の類似度閾値を示す)

類似度尺度	weight	measure	F 値							
			タスク I				タスク II			
			新聞		Web		新聞		Web	
P-S	P_β	SIMPSON	0.985	(0.13)	0.973	(0.13)	0.807	(0.19)	0.876	(0.17)
P-J	P_β	JACCARD	0.981	(0.04)	0.945	(0.03)	0.743	(0.04)	0.805	(0.02)
P-SJ	P_β	SIMPSON-JACCARD	**0.988**	(0.08)	**0.975**	(0.08)	**0.817**	(0.13)	**0.878**	(0.11)
Lin98[4]	MI	LIN98	0.985	(0.08)	0.949	(0.06)	0.748	(0.10)	0.805	(0.06)
Lin02[5]	MI′	COSINE	0.984	(0.14)	0.955	(0.13)	0.758	(0.16)	0.818	(0.12)
相澤 [6]			0.982		0.971		0.752		0.862	

とがわかる。表 4 には比較のために相澤 [6] の手法の精度も記載した。相澤は、新聞から作ったデータに対しては 31 年分の新聞テキストから、Web から作ったデータに対しては約 4,000 万 Web ページから、分布類似度を計算している。また、類似度尺度としてはタスク I の新聞には P-J を、他のタスクには P-S を利用している。P-SJ はすべての評価セットにおいて相澤の手法を上回っている。

次に、コーパスサイズ、曖昧性のある係り受けの除外の効果、語の単位、共起要素の格について実験を行なった[4]。

コーパスサイズ　コーパスサイズを変更することにより、分布類似度の精度がどのように変わるかを調べた。ここでは共起ベクトル (ii) を利用した。結果を図 3 に示す。すべての評価セットにおいて、コーパスサイズを大きくするにつれて精度が向上していることがわかる。また、精度が飽和していることから我々が用いたコーパスのサイズで十分であることがわかる。

曖昧性のある係り受けの除外　2 節で述べた、曖昧性のある係り受けを除外することによる効果を調べた。この実験では類似度尺度 P-SJ を用いた。図 4 に結果を示す。コーパスサイズが小さい時は曖昧性のある係り受けを捨てた方が精度が悪くなっている。これは係り受け誤りによる誤った共起要素が追加されるよりも、データスパースネスが解消される方が大きいと考えられる。コーパスサイズが大きくなると、精度が同等、もしくは少しよくなる。たとえ精度が同じであっても表 3 に示したとおり、曖昧性のある係り受けを捨てることによって、共起要素が減り、分布類似度の計算時間が速くなる。

語の単位　語の単位として、単名詞と複合名詞を利用した場合の精度を比較した。この実験では類似度尺度として P-SJ を用いた。図 5 に結果を示す。コーパスサイズが小さい場合はデータスパースネスの問題のため、複合名詞を利用すると精度が低下しているが、コーパスサイズが大きくなると精度が少し向上している。これにより我々が実験に用いたサイズは複合名詞の分布類似度を計算するのに十分であることが示された。複合名詞を利用することによって改善した例として、「神経衰弱」と「ゲーム」(類義語) がある。単名詞に汎化した場合、「衰弱」と「ゲーム」の類似度を計算することになり、類似度が下がるが、複合名詞を単位とした場合は類似度が高くなり、正解となった。

共起要素の格　共起要素として利用した格のうち、どの格が有効かを調べた。この実験では共起ベクトル (i) を用いた。すべての格から、ガ、ヲ、ニ、カラ、ト、ヘ、マデ、ヨリ、ノを一つずつ除いて精度を出した。結果を表 5 に示す。「ノ」を除外した場合が最も精度が低下したことより、「ノ」が有効であることがわかる。また、予備実験により「デ」を追加すると精度が下がったので利用しなかった。これは「デ」格は様々な用法で利用されるためだと思われる。

4.4 誤り分析

タスク I は自動生成されたデータであるので、類義語かどうかの判断が微妙なものが存在する。例えば、

[4]紙面のスペースの都合上、ウェブテキストから作成された評価セットの結果のみを記載している。

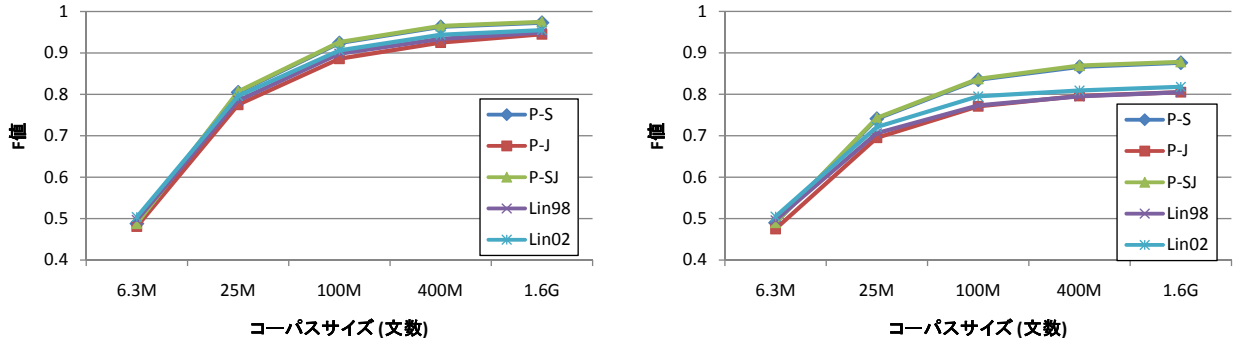

図 3: コーパスサイズと F 値の関係 (左: タスク I, 右: タスク II)

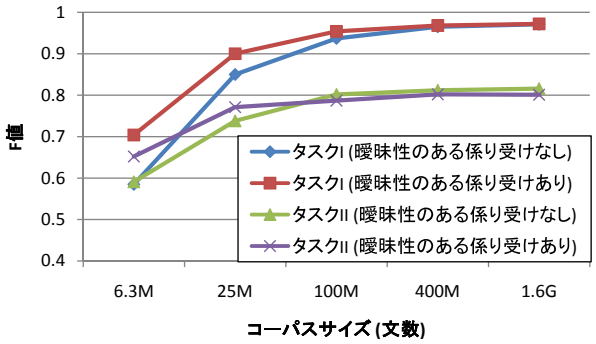

図 4: 曖昧性のある係り受けの除外の効果

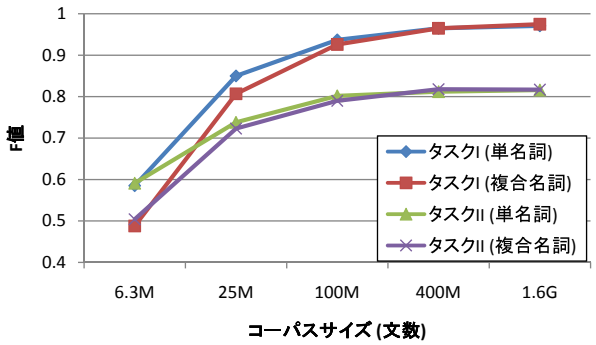

図 5: 語の単位 (単名詞と複合名詞)

パターン「銀行 (A) や空港 (B) などの場所 (C)」から「銀行」と「空港」が類義語となっているが、システムは非類義語と判定してしまっている。

他の問題として多義性の問題がある。評価セットに含まれている語が多義であれば、本来示すべき値よりも類似度が下がる傾向にある。例えば、「マウス」と「豚」は評価セットにおいて類義語となっているが、システムは非類義語と判定してしまう。「マウス」は「ニ:注射」、「ノ:胎児」といった共起要素だけでなく、「デ:ドラッグ」、「ノ:ホイール」といった共起要素ももつ。その結果、「豚」との類似度が下がってしまう。この問題には語の多義性解消を行なうことにより対処する予定である。

表 5: 共起要素の格と F 値の関係 (括弧内は最も精度がよい時の閾値を示す)

格	F 値			
	タスク I		タスク II	
― ガ	0.971	(0.07)	0.870	(0.10)
― ヲ	0.970	(0.07)	0.873	(0.10)
― ニ	0.971	(0.07)	0.871	(0.11)
― カラ	0.971	(0.08)	0.873	(0.10)
― ト	0.971	(0.07)	0.872	(0.10)
― ヘ	0.971	(0.08)	0.872	(0.10)
― マデ	0.971	(0.08)	0.872	(0.10)
― ヨリ	0.971	(0.08)	0.872	(0.10)
― ノ	0.966	(0.08)	0.867	(0.11)
すべて	**0.971**	(0.07)	**0.872**	(0.10)
＋ デ	0.971	(0.07)	0.870	(0.11)

5 おわりに

本稿では超大規模ウェブコーパスを用いて分布類似度を計算する手法について述べた。コーパスサイズを大きくするにつれて精度が向上することを示した。

今後は分布類似度を用いて大規模格フレームを構築する予定である。

参考文献

[1] James Richard Curran. *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh. College of Science, 2004.

[2] J.R.Firth. *Studies in Linguistic Analysis*, chapter A synopsis of linguistic theory. Oxford, 1957.

[3] Daisuke Kawahara and Sadao Kurohashi. Japanese case frame construction by coupling the verb and its closest case component. In *Proceedings of HLT 2001*, pp. 204–210, 2001.

[4] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*, pp. 768–774, 1998.

[5] Dekang Lin and Patrick Pantel. Concept discovery from text. In *Proceedings of Conference on Computational Linguistics (COLING 2002)*, pp. 577–583, 2002.

[6] 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.