検索エンジンを用いた関連語の自動抽出

Automatic Extraction of Related Terms using Web Search Engines

渡部 啓吾 †， Danushka Bollegala †， 松尾 豊 ††， 石塚 満 †

† 東京大学大学院情報理工学系研究科　†† 東京大学大学院工学系研究科
東京都文京区本郷 7-3-1

Keigo WATANABE †　Danushka BOLLEGALA †　Yutaka MATSUO ††　and　Mitsuru ISHIZUKA †
† Graduate School of Information Science and Technology, The University of Tokyo　†† School of Engineering, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan

要約

クエリー拡張や類似検索など，さまざまな情報検索のタスクにおいて，関連語が登録されているシソーラスは必要不可欠な言語資源である．人手で作られたシソーラスである WordNet やロジェのシソーラスを使っている情報システムは多数存在するが，関連語シソーラスを人手で構築または更新する作業は大変コストがかかるだけでなく，新語や既存の単語の新たな使い方をカバーできないという問題がある．本論文ではウェブを膨大なテキストコーパスとみなし，検索エンジンを通して関連語を抽出するための手法を提案する．提案手法では，ウェブ検索エンジンから得られるスニペットを用い，効率良く関連語を抽出することができる．

キーワード ： ウェブマイニング，情報抽出，関連語抽出, 検索エンジン

Abstract

Semantic lexicons, such as Roget's Thesaurus or WordNet, act as useful knowledge resources in natural language processing applications. However, such manually created lexical resources do not always reflect the new terms and named entities frequently found in the Web. Moreover, manually maintaining lexical resources are costly and time consuming. Motivated by this challenge, we propose a method to automatically extract related terms using the web as a corpus. The proposed method exploits snippets retrieved from a web search engine and efficiently finds related terms.

Key words ： Web mining, information extraction, related terms, search engines

1 はじめに

関連語とは，ある語に対して関連の強い語を指し，関連性のタイプによって分類することができる．例えば，名詞同士の関係に絞ってみると，Miller らが表 1 で示しているような同義，対義，上位–下位，全体–部分という 4 つの関係が定義できる [3]．ある語に対する関連語を用いて，検索時のクエリ拡張や語の曖昧性解消など，さまざまなタスクに応用することができる．例えば，情報検索時に，ユーザが入力したクエリに対する関連語の情報を利用することで，提供する情報の幅を広げることができる．他にも，一語で複数の意味を持つ語が出現したときに，それぞれの語義に対する関連語の情報を持っていれば，周辺の語群から対象とする語の意味を推定することが可能となる．

シソーラスやオントロジーは，言語処理を行う上で利用価値の高い知識ベースであるが，これを人手で構築し，維持していくためには多くの手間と時間がかかる．関連語を自動的に抽出することができれば，そのコストを減らすことが可能である．したがって，関連語の自動抽出は実用的な重要性の高いタスクである．

関連語を抽出する際に必要になるのが語同士の関連の強さであり，その測り方としては，2 語の直接的な共起を用いる手法と，周辺の語の類似性を利用する手法の 2 つに大きく分けられる．前者は，語同士が共起した回数や，共起した際の周辺の語（パターンと呼ぶ）を用いて，その関連度を測定する手法である．例えば，ある語 ‘X’ と ‘Y’ が，“X is a Y” というパターンで出現することが多ければ多いほど，‘X’ と ‘Y’ の is-a 関係が強いとみなすことができる．後者は，出現した際の文脈が類似している語同士は関連度が高いとする手法で，同義語や類義語の抽出に利用できる．例えば，“car” や “automobile” という語の周辺には，“drive” や “rental” という語が多く出現しているので，“car” と “automobile” は類似しているとみなすことができる．本論文では，同義関係や類義関係に限らず多様な関係を取得するため，前者の手法を用いる．

これまでの関連語抽出の研究では，新聞や雑誌などをコーパスとして用いることが多く [6, 10]，新しい語や意味の変化に対応できない問題があった．一方で，近年のウェブの急速な発展に伴い，ウェブからの情報抽出の研究が盛んに行われている．関連語を抽出する際にウェブをコーパスとして用いることで，新しい語や意味の変化に随時対応できるだけでなく，100 億以上のウェブページを対象として，より広い領域からの関連語の抽出が可能となる．また,「人の名前」と「勤めている会社」のような，時間とともに変化する関係についても取得することができる．

ウェブ上のデータ量は膨大であるため，ウェブをコーパスとして利用するためには，検索エンジンを使うことが有効である．検索エンジンをインタフェースとして用い，ウェブから関連語の抽出を行う際には解決しなければならないいくつかの課題がある．まず，与えられたクエリーに対し，ウェブ検索エンジンが返すことのできる検索結果が制限されていることが多い．例えば，Google の場合，1 つのクエリーに対してユーザが閲覧できるのは上位 1000 件の検索結果のみである．こういった制約は，通常，検索エンジンは検索結果を瞬時に処理する必要があり，多くのユーザは上位の数件しか結果を見ないことに由来する．新聞記事等のコーパス全体を利用する手法に比べ，検索エンジンを使った関連語の抽出では，この制約の範囲内で工夫して処理を行う必要がある．

さらに，多くの検索エンジンはクエリーとページの関連度を用い，検索結果を順位付けしているので，上位の結果だけでは，検索エンジンが返すページに結果が依存する．そのため，検索エンジンが返す検索結果に関する URL を全てダウンロードし，テキストの処理を行うことも考えられるが，通常，URL を全てダウンロードするには長い時間が必要となり，処理効率の面では望ましくない．多くの検索エンジンでは，検索結果と同時にクエリーとして入力された語が出現する文脈を短い要約（スニペットと呼ぶ）として表示するため，これを利用することを考える．スニペットを用いることによって，検索結果の URL をダウンロードする手間が省略でき，短い時間で処理が可能である．しかし，スニペットは断片的なテキストであることが多く，完全な文を入力前提とする係り受け解析など高度な言語処理に向いていないため，短いテキストの中で適用可能なパターン抽出の手法を用いる必要がある．

本論文では，検索エンジンを利用することでウェブ全体をコーパスとし，スニペットを対象としたパターンによって関連語を抽出する手法を提案する．具体的には，あらかじめ既存の辞書（例えば WordNet[3]，ロジェのシソーラス [12]）などを用いて，求めたい関係 R（例えば同義関係）の正例となるデータを作り，関係の生じるパターンを学習させておくことで，ある単語 W（例えば car）が与えられたとき，W と関係 R に当たる語（例: automobile）の順位付けされたリストを獲得することができる．

本論文は以下のように構成される．まず 2 章で関連研究について述べ，3 章で提案手法について詳しく説明する．続いて 4 章で評価実験を行い，5 章でまとめを行う．

表 1　名詞間の関係の種類とその例

関係の種類	関連語の種類	意味	例
Synonymy（同義）	Synonym（同義語）	意味がほぼ同じ語	car - automobile
Antonymy（対義）	Antonym（対義語）	意味が反対の語	rise - fall
Hyponymy（上位－下位）	Hypernym（上位語）	上位概念を表す語	dog - animal
	Hyponym（下位語）	下位概念を表す語	dog - poodle
Meronymy（全体－部分）	Holonym	部分に対する全体を表す語	dog - canis（イヌ属）
	Meronym	全体に対する一部を表す語	dog - flag（尻尾）

2　関連研究

2.1　関連語の抽出

名詞同士の関係にはいくつかの種類があり，Millerらは表 1 のような 4 つの関係を定義している [3]．これらの関係のうち，Hearst[6]，Pantel ら [10]，Snow ら [13] は上位－下位関係，Lin ら [2]，榊ら [14] は同義関係，Girju ら [5] は全体－部分関係について関連語の抽出を行っている．

パターンを用いた関連語の抽出で有名な研究のひとつに，Hearst による研究がある．Hearst は人手で式 1，式 2 などの信頼できるパターンをいくつか作り，文書内にそのパターンが生じる部分から上位－下位関係にある語を抽出する手法を提案した [6]．

$$NP_0\ such\ as\ NP_1,\ NP_2 \ldots, (and \mid or) NP_n \quad (1)$$

$$NP_1, \ldots, NP_n (and \mid or) other\ NP_0 \quad (2)$$

しかし，表 1 に示すように関連語の間では上位ー下位関係以外にもさまざまな関係が存在するので，人手で作成したパターンのみで全ての関連語を抽出するのは難しい．特に，ウェブのようにさまざまなドメインのテキストが含まれる多様なコーパスの場合は，それぞれのドメインでどのようなパターンが使われやすいかを事前に把握することができないため，本論文の提案手法のような抽出パターンの学習が有効となる．

Pantel ら [10] は，上位－下位関係にある二語が共起する文を抽出し，品詞のタグ付け（Part-of-Speech Tagging）を行ったあと，二語間を結ぶ文脈を抽出パターンとして抽出する方法を提案した．選択されたパターンを再びコーパス内の文にマッチさせ，上位ー下位語関係にある単語ペアを抽出する．このサイクルを繰り返し，ブートストラップを行うことによって抽出パターンと関係語のペアを増やす．しかし，こういったブートストラップ的な手法では，意味ドリフト（semantic drift）と呼ばれる現象が生じることが知られており，サイクルが増えると雑音（すなわち信頼性の低い抽出パターンや関連性の薄い単語）が含まれるようになり，それ以降のサイクルがその雑音に大きく影響されるという問題がある．本提案手法では複数の単語ペアを用い，信頼度の高いパターンを用いるため，雑音による意味ドリフトの影響を受けにくいという特長がある．

2.2　ウェブ検索エンジンを用いた情報抽出

ウェブからの情報抽出においては検索エンジンは大きな役割を果たす．これまで，ウェブからの情報抽出や知識抽出の研究では，検索エンジンの提供するいくつかの機能のうち，ヒット件数を利用することが多かった．ある語を検索したときのヒット件数はその語の出現頻度を表し，二語を組み合わせて検索したときのヒット件数は二語が共起する頻度を表すため，基本的な語の出現の統計情報を取得することができ，語の関連度を計ることができる [14]．

しかし，ヒット件数だけでは語がどのように出現したかという文脈を把握することができないため，Bollegala らはヒット件数に加え，検索結果のスニペットを用い，そこに生じたパターンも利用して関連度を計る手法を提案した [1]．Cimiano らは検索エンジンのスニペットを利用して，語を定義する 4 つの要素を抽出した [9]．検索エンジンの提供する機能にはいくつかの種類があり，その特徴も異なるので，目的に合わせて情報を取捨選択することが重要である．最も単純に考えると，検索によって得られたページをダウンロードして，ページのテキストを取得し，それを解析する手法が考えられるが，ページを全てダウンロードするには時間がかかる．本研究で関連語の抽出に利用する部分は，語が出現する前後の文のみであるため，スニペットだけでもその役割は果たせると考えられる．提案手法では，実際に抽出するための効率を重視し，検索結果のスニペットを利用して，パターンの抽出や関連語の抽出を行う．

3　手法

ウェブ上には新聞や雑誌などに比べ整形されていない文章が多いため，必然的に関連語の抽出の精度が鍵

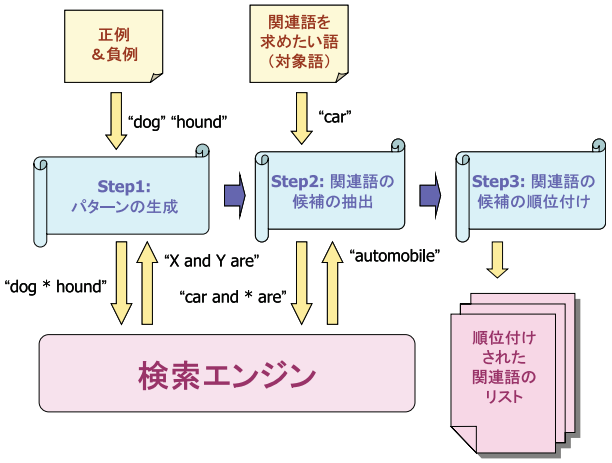


図 1　全体の流れ

となる．まして，スニペットという短文を利用する本手法では，信頼度の高いパターンを得るための方法が鍵となる．そこで，本論文では以下のように 3 段階の手法を提案する．

1. ステップ 1 : 正例となるデータからパターンを生成する．
2. ステップ 2 : パターンから関連語の候補を抽出する．
3. ステップ 3 : 関連語の候補の順位付けを行う．

提案手法の概要を図 1 に示す．以下では，各ステップについて説明する．

3.1　ステップ 1: パターンの生成

パターンを作成するにあたって，検索クエリとしてワイルドカード ‘∗’ を使うため[1]，それに対応した検索エンジンを用いる必要がある．本研究では，以下 Google（`http://google.com/`）を用いる．

まず学習データとして，正例（R という関係にある語のペア）と負例（R という関係にない語のペア）のリストを用意する．このとき R は任意であり，抽出したいものに応じた関係を指定する．抽出したい関係 R に対して正例と負例となる単語ペアを選択するための具体な方法を 4 節で説明する．

次に，得られたリストから 2 つの語の間に ‘∗’ を k 個入れたクエリを用い，検索を行う．予備実験により，関連語の抽出に有用なパターンは，語のあいだの単語数が 3 以下のものがほとんどであったため，本研究では，$k = 1, 2, 3$ とする．すなわち，二つの語を ‘X’, ‘Y’ とすると，“X∗Y”，“X∗∗Y”，“X∗∗∗Y”，“Y∗X”，“Y∗∗X”，“Y∗∗∗X” の 6 つを利用する．得られたスニペットの上位 M 件を利用し，2 つの語が共起するときに現れるパターンを以下の手順で抽出する．

1. 単語ペア (X,Y) を含む上記の 6 つの検索クエリから得られるスニペット中の X と Y の全ての出現を変数 **X** と **Y** で置き換える．
2. あるスニペット中に **X** より前に出現する部分を prefix と言い、**X** と **Y** の間に出現する部分を midfix と言い、**Y** より後に出現する部分を suffix と言う．尚、prefix、midfix、suffix の長さはそれぞれ n_{pre}, n_{in}, n_{post} 単語以下とする．これらのパラメータの値を調整することによって X と Y が共起する文脈の幅 (context window) を変更することができる[2]．
3. **X** と **Y** 両方を含みかつ (2) の条件を満たす全ての部分列を語彙パターンとして抽出する．

例えば，“dog∗∗animal” という検索を行い，“...no *dog or other* animal shall be left...” というスニペットが得られたとすると，そこから抽出されるパターンは表 2 と同じものとなる．

このように網羅的にパターンを抽出すると，その量は膨大なものになるので，得られたパターンの有意さを計る指標が必要となる．本論文では，あるデータ集合内での統計的な偏りを表す指標である χ^2 値を用いる．具体的には，パターン v それぞれについて表 3 のように分割表を作成し，式 3 に従って計算する．このように χ^2 値は出現頻度の正規化項を持ち，あるパターンの正例における出現確率と負例における出現確

表 2　パターン抽出の例

検索クエリ	“dog∗∗animal”
スニペット	...no *dog* or other *animal* shall be left...
抽出するパターン	no X or other Y no X or other Y shall no X or other Y shall be X or other Y X or other Y shall X or other Y shall be

[1] 対応した検索エンジンであれば，‘∗’ の部分はどのような語でもマッチするような検索を行うことができる．英語では ‘∗’ 一つが一単語にマッチする．

[2] 予備実験において，単語数を制限せず生成したパターン上位 100 件には，前に二単語，間に四単語，後に二単語以下のものしかなかったため，本研究では，$n_{pre} = 2$, $n_{in} = 3$, $n_{post} = 2$ と設定した．また Google では記号を用いた検索ができないため，記号は無視する

表 3　χ^2 検定の分割表

出現頻度	パターン v	パターン v 以外	合計
正例	p_v	$P - p_v$	P
負例	n_v	$N - n_v$	N

率が異なるほど大きな値となるので，χ^2 値が大きいパターンほど関連語を抽出するために有用である．

$$\chi^2 = \frac{(P+N)(p_v(N-n_v) - n_v(P-p_v))^2}{PN(p_v+n_v)(P+N-p_v-n_v)} \quad (3)$$

実際に得られた全てのパターンに対して χ^2 値を計算し，その値によって重み付けを行う．ただし，式 3 中の $p_v(N-n_v) - n_v(P-p_v)$（分子の 2 乗される中身）の値が負になるときは，負例のほうに偏って出現するパターンであるためそのようなパターンを利用しない．ステップ 2 以降では，χ^2 値が上位 N 件のパターンを利用する．得られたパターンの一部を表 4 に示す．

3.2　ステップ 2: 関連語の候補の抽出

ステップ 1 により，ある関係を見つけ出す上で重要となるパターンを抽出する．次にステップ 2 では，検索エンジンを利用して，関連語を求めたい語（以下，対象語と呼ぶ）と得られたパターンがウェブ上で共起する文を探し出し，そこから対象語の関連語の候補を抽出する．ここで，検索結果を得るためのクエリはいくつかの種類が考えられ，最も単純には対象語をそのままクエリとする手法が考えられる．しかし，検索エンジンではヒット件数と同じ数のスニペットを参照できるわけではなく，参照できる数は大きく制限されるため，候補語を抽出する上で充分なスニペットを得ることができない．そこで，本論文では候補語とそれぞれのパターンを組み合わせてクエリとすることで，パターンに対応したクエリで検索を行い，候補語を抽出するのに充分な量の検索結果を取得する．

具体的には，まずステップ 1 によって得られたパターンの ‘X’ と ‘Y’ の部分を，対象語と ‘∗’ にそれぞれ置き換えてクエリを作成する．例えば，“X and Y are” の関連において、“dog” の関連語を求めたければ，“dog and ∗ are”，“∗ and dog are” のような 2 種類のクエリが得られる．そのクエリを用いて検索し，得られた上位 M 件のスニペットから，パターンの ‘∗’ にあたる部分から n グラム[3]を抽出し，関連語の候補とする．この際，出現頻度の高い極めて一般的な語（例: the, or, of...）はストップワード[4]として除く．また，PorterStemmer[8] によりステミングを行い，ステミングの結果が同じとなる語（例: dog と dogs）は類似候補としてひとつにまとめる．

3.3　ステップ 3: 関連語の候補の順位付け

ステップ 2 で得られた関連語の候補は，ウェブ上から取得するという性質上，膨大な量の候補が抽出されることがある．その中には対象語に関連する語も含まれるが，同時に，関連が少ない，もしくは全く関連性のない語も多く含まれている．対象語の関連語のみを精度良く抽出するためには，それぞれの候補について，関連語としての適切さをスコア付けし，順位付けをすることが必要である．

ここでは，ステップ 1 で得られたそれぞれのパターンごとの出現頻度を用いて，関連語のスコア付けを行う．指標としてはさまざまなものが考えられるが，最も単純に考えると，各パターンに該当するかしないかを 0-1 で表し，該当するパターンの数を足し合わせたものを指標 PF(c)(式 4) とする．

$$\mathrm{PF}(c) = \sum_v f(c_v) \quad (4)$$

ただし，$f(c_v) = |\{v|c_v > 0\}|$ とする．また，c はステップ 2 で得た候補語，v はあるパターンとし，候補語ごとのパターンの出現頻度を c_v とする．上記の指標は，すなわち候補語 c ごとに該当するパターンの種類数を求めていることになる．

次に，各パターンにいくつかの候補語があるかという頻度情報を考慮し、指標 TF(c)(式 5) を計算する．

$$\mathrm{TF}(c) = \sum_v c_v \quad (5)$$

例えば，ひとつのパターンにしか該当しないが，そのパターンで何件ものページが該当するものは高いスコアであると考える．

さらに指標を詳細にすることを考える．ステップ 1 で χ^2 値によって選別されたパターンの中でも，候補語の抽出精度が高いものもあれば，そうでないものもある．例えば，“X synonyms Y” というパターンは χ^2 値はそれほど高くないが，精度が高い．一方，“X and Y” は χ^2 値は高いが精度は低い．こういったパターンごとの精度の違いを考慮するために，パターンごとに重みを計算する．この重みを用いて，上記の $PF(c)$, $TF(c)$ の値を WeightPF(c)(式 6) と WeightTF(c)(式 7) として改良する．

[3] 本論文では $n = 1$ もしくは 2 と設定した．

[4] 本研究では一般的なものを利用した．http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/

$$\mathrm{WeightPF}(c) = \sum_v weight_v \times f(c_v) \qquad (6)$$

$$\mathrm{WeightTF}(c) = \sum_v weight_v \times c_v \qquad (7)$$

ただし，$weight_v$（パターン v の重み）には，それぞれのパターンによって実際に関連語を取得し，その F 値を用いた．F 値 (式 10) とは適合率 (式 8) と再現率 (式 9) の調和平均を取ったものであり，それぞれが次のように計算できる．

$$\text{適合率} = \frac{\text{候補中の正しい関連語の数}}{\text{関連語の候補の数}} \qquad (8)$$

$$\text{再現率} = \frac{\text{候補中の正しい関連語の数}}{\text{正しい関連語の数}} \qquad (9)$$

$$F\text{ 値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \qquad (10)$$

パターンの重み ($weight_v$) を計算するための開発データとしていくつかの単語に関する関連語をロジェのシソーラスから人手で選択した．ただし，4 節で使われる評価用のデータに含まれている単語とその単語の関連語が開発データからあらかじめ除外してある．F 値による上位パターン数件を表 4 に示す．

上記の議論によると、パターンの重みとして PF(式 4),TF(式 5),WeightPF(式 6) と WeightTF(式 7) の 4 つの手法が考えられるが、PF は出現したパターンの種類数を表す最も粗い指標であり、TF、WeightPF、パターンごとの重みと出現頻度の積の総和である WeightTF の順により細かい重みを考慮した指標となる．そこで，以下の実験では，この 4 つの中でも，最も粗い指標である PF と最も細かい指標である WeightTF を用いて評価する．TF と PF はそれぞれ WeightTF と WeightPF で $weight_v = 1$（つまり、全ての語彙パターンは目的とする関係を同じ精度で表している）を仮定した WeightTF と WeightPF の特殊な場合として解釈することができる．

4 評価

提案手法は正例となる語のペアをクエリとし，生じたパターンを学習して関連語を抽出する手法であり，言語や品詞，関係の種類によらない．ここでは英語の名詞を対象とし，その中でも代表的な同義関係と上位－下位関係に関して評価実験を行う．

抽出パターンを学習するための正例と負例となる単語ペアを WordNet から次のように選択する．まず，同義関係に関する正例を作成するために WordNet で名詞として登録されている単語をランダムに選択し，その synset の中から同義語を一つランダムに選択する．WordNet では単語の語義ごとに同義語が登録されており，ある単語の同義語となるものの集合は synset と呼ばれる．その操作を繰り返すことによって同義関係の正例となる 1000 個の単語ペアを抽出する．次に，同義関係の負例を選択するために，WordNet で名詞として登録されている 2 つの単語をランダムに選択し，更に選択されたその 2 つの単語は WordNet 中の全ての単語の synset の中にも含まれていないことを確認する．この操作を繰り返すことによって同義関係の負例となる 1000 個の単語ペアを抽出する．上位-下位関係についても同じようにして正例と負例となる単語ペアを 1000 個ずつ抽出する．その際には synset ではなく，hypernym set として登録されている単語の中から選択する．なお，全ての実験においてパラメータは $M = 100$ (上位何件のスニペットを用いるか)，$N = 100$（上位何件のパターンを用いるか）に固定した．

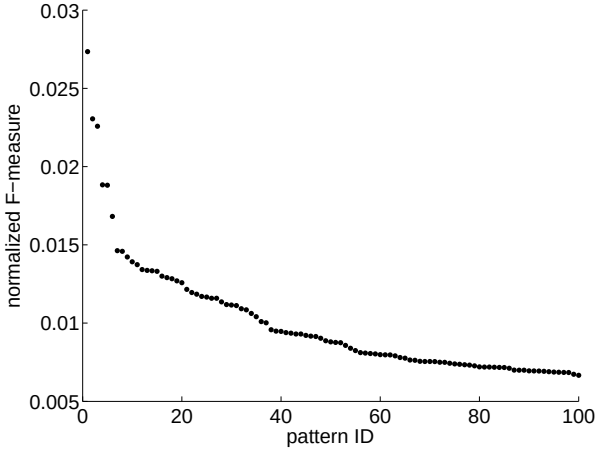


図 2　同義語関係に関して抽出したパターンの重みの分布

同義語関係、上位語関係、下位語関係について抽出したパターンに関してそれらの F 値と χ^2 値を表 4 に示している．それぞれの関係について合計 100 の語彙パターンを抽出したが、スペースの都合上表 4 では F 値の上位 10 個のパターンのみを表示している．表 4 を見ると同義語関係、上位語関係、下位語関係を表す様々な語彙パターンが提案手法によって抽出されていることが分かる．更に、図 2 では同義語関係に関して抽出した全 100 個のパターンの重み、$weight_v$、の分布を示す．上記 3.3 節で説明された通り $weight_v$ は式 10 によって語彙パターンの F 値として計算されており、図 2 の横軸はその F 値によって順序つけた場合のランクである．尚、図 2 の重み分布はその面積が 1 になるように全語彙パターンの重みの総和で割ることで

表 4　重み付けされたパターンリスト（F 値上位 10 件）

同義語抽出 X,Y:同義語			上位語抽出 X:下位語			下位語抽出 X:上位語		
パターン	F 値	χ^2 値	パターン	F 値	χ^2 値	パターン	F 値	χ^2 値
synset X Y	0.2736	68.4	X is a ∗ who	0.2996	28.2	∗ or other X	0.1569	73.2
X synonyms Y	0.2051	50.8	X by ∗ of	0.1377	20.2	a ∗ is a X	0.1346	8.19
syn X Y	0.1876	64.7	X a ∗ who	0.1259	44.2	X such as a ∗	0.1241	42.6
an X or Y	0.1383	29.9	X or other ∗	0.1082	73.6	X or ∗ as	0.1011	15.7
X or Y a	0.1128	20.9	a ∗ or X	0.1031	43.3	∗ n the X of	0.0933	17.6
X or Y was	0.0978	12.8	∗ and X for	0.0942	25.4	X or ∗ in	0.0907	15.7
X or Y is	0.0900	79.0	∗ or X that	0.0886	20.7	X called a ∗	0.0891	24.2
X or Y in	0.0747	43.1	the ∗ or X	0.0849	57.8	X or ∗ is	0.0861	18.4
a X or Y	0.0745	128.1	∗ or X of the	0.0815	11.7	X or ∗ of the	0.0807	16.5
as a X or Y	0.0692	13.5	X a ∗ that	0.0763	44.0	X by ∗ of	0.0801	32.1

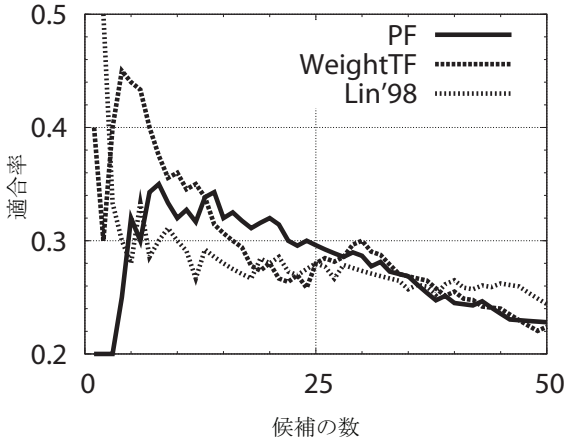


図 3　Lin'98 との比較

正規化してある．図 2 を見ると、パターンの重み分布はロングテールとなっていることが分かる．これは上位少数個の語彙パターンだけではなく、多数の語彙パターンによって同義語が抽出されていることを意味する．したがって、人手で作成した少数の語彙パターンだけでは抽出できないような同義語も提案手法によって抽出可能となる．なお、上位語関係と下位語関係についてもそれらのパターン分布に関してこのロングテール現象が観察できた．

4.1　関連研究との比較

関連語自動抽出の代表的な先行研究としては，Lin[2)] によるシソーラスの自動構築手法がある．本節では，関連語抽出の先行研究との定量的比較として Lin の手法と比較を行う．Lin は，構文解析された新聞記事などの文章を利用して，自動的にシソーラスを構築する手法を提案しており，ある語に対して他の語の依存関係が似ている度合いを評価して順位付けする．関連語抽出に関する論文では比較手法としてよく用いられるため，本手法でも比較手法として用いる．比較実験では {cord, forest, fruit, glass, slave} という一般的な 5 語を対象語とし，本論文の提案手法で抽出した同義語上位 50 件と，Lin のシソーラスに登録されている同義語上位 50 件について平均の適合率を調べた．

表 5　Lin'98 との平均の適合率の比較

候補の数（上位 n 件）	@5	@10	@20	@30
Lin'98	0.28	0.30	0.28	0.27
PF	0.32	0.32	0.32	0.29
WeightTF	0.44	0.36	0.28	0.30

その結果を表 5 に示す．表 5 を見ると，上位 5-30 件の適合率を比較した場合では提案手法にやや優位な点が見られるが，全体的にはほぼ同程度の適合率となっており，提案手法はうまく関連語を抽出できていることが分かる．なお，提案手法は，Lin の手法のようにあらかじめコーパスを用意したり構文解析をする必要がなく，同程度の適合率であっても本手法の有用性は高いと言うことができる．

4.2　既存のシソーラスとの比較

次に，既存のシソーラスとの比較を行う．表 5 の 5 単語を対象とし，正解データは Roget's Thesaurus (同義語)，WordNet（上位語，下位語）に関連語として登録されているものとした．

図 4-図 6 に，それぞれの抽出方法を用いた際の「取得した語数/正解となる語数」に対する平均の F 値を示す．同義語の場合には，F 値は最高で 0.158

（WeightTF），上位語の場合には 0.114（PF），下位語の場合には 0.084（WeightTF）となった．いずれの場合にも，取得語数が 0 に近いときは F 値は低い値だが，取得語数が増えると精度が高くなり，さらに多くなると精度が収束，もしくは下降する．同義語や下位語の場合には，「取得した語数/正解となる語数」= 1 近辺，すなわち，正解として登録されている語数と同程度の語数を取得したときが最も F 値が高くなったが，上位語の場合にはより多くの語数を取得した方が F 値が高くなっている．したがって，「取得した語数/正解となる語数」に関する適切な閾値は，求めたい関係によって異なることが分かるが，概ね 1〜1.5 程度が適切であると考えられる．

4.3 候補語の順位付けに用いる指標の比較

最後に，提案手法で用いた指標の優位性を示すため，ウェブのヒット件数を用いた一般的な指標との比較を行う．Bollegala らは，一般的な指標である Jaccard 係数，Overlap 係数，Dice 係数，PMI（相互情報量）を元にし，ウェブの検索エンジンによるヒット件数を利用した $WebJaccard$，$WebOverlap$，$WebDice$，$WebPMI$ という指標を式 11-14 のように定義した [1]．ここで，$H(P)$ は P というクエリで検索したときのヒット件数，$H(P \cap Q)$ は P AND Q というクエリで P と Q の二語を AND 検索したときのヒット件数とする．また，$WebPMI$ における N は，確率の定義から検索の対象となるウェブページの総数であるが，正確な値は知ることができないので，本論文では Google によってクロールされている 10^{10} ページとした．

$$\mathrm{WebJaccard}(P,Q) = \frac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)} \tag{11}$$

$$\mathrm{WebOverlap}(P,Q) = \frac{H(P \cap Q)}{\min(H(P), H(Q))} \tag{12}$$

$$\mathrm{WebDice}(P,Q) = \frac{2H(P \cap Q)}{H(P) + H(Q)} \tag{13}$$

$$\mathrm{WebPMI}(P,Q) = \log_2 \left(\frac{\frac{H(P \cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}} \right) \tag{14}$$

以上，4 つの既存指標に提案手法である 2 つの指標を加えた，計 6 つの指標について比較した．ここでは，同義語を豊富に持つ一般的な語として “magician” を選び[5]，ロジェのシソーラスに登録されている 88 語を正解データとし，指標別に取得したデータの個数に対する F 値の推移を図 7 で示した．

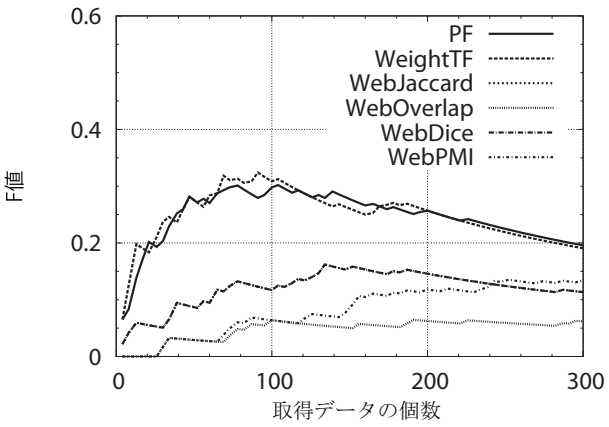


図 7　順位付けに用いる指標の比較

既存指標のなかで最も F 値が高かったのは WebDice，WebJaccard の 2 つであり（この 2 つは，グラフ中でほぼ重なっている．），取得語数が 134 のときに，F 値が最高で 0.162 であった．他の WebPMI，WebOverlap はこれよりも悪い結果であった．他の文献 [4, 15] では，WebPMI や WebOverlap が有効であるとされているが，本研究のように，あらかじめ抽出する関係を定めた上での関連語の抽出においては，異なる結果となった．

提案手法である PF，WeightTF では，取得語数が 90〜100 のときに F 値が高く，最高で 0.324 となり，WebDice や WebJaccard を大きく上回り，2 倍の値となった．PF，WeightTF のいずれも，ほとんど F 値に優劣はないが，取得語数が 120 以下の領域では，やや WeightTF が上回っている．この 2 つの手法は，取得語数が増えると，徐々に精度が落ちているが，これは網羅性が高くなる結果，Recall が上がり Precision が下がって結果的に F 値が下がるためである．また，取得データ数が 30-50 と小さい領域でも，他の手法に比べて良い精度を実現しており，正解の語を上位にランキングできていることが分かる．

以上の実験結果から，提案手法は既存手法と比べて，F 値を大きく改善することができた．すなわち，一般的な関連性ではなく，同義語や上位語，下位語という特定の関係であれば，提案手法のようにパターンを利用する効果が高い．

実際に “magician” という語に対して，提案手法に

[5] 既存の指標では，検索エンジンに多くのクエリを入力する必要があり，クエリ数の制限から多くの語に対して評価実験を行うことができなかった．予備実験により本手法の一般的な傾向が出やすい語に定めた．

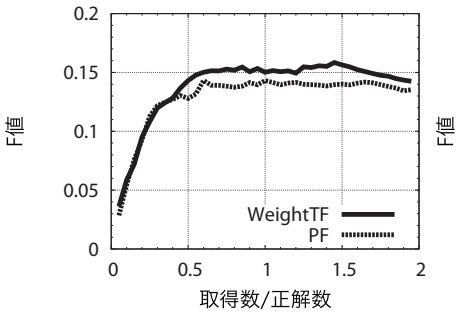


図 4　同義語抽出

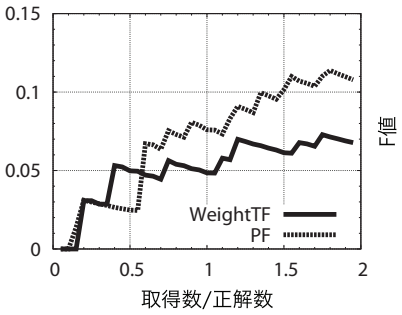


図 5　上位語抽出

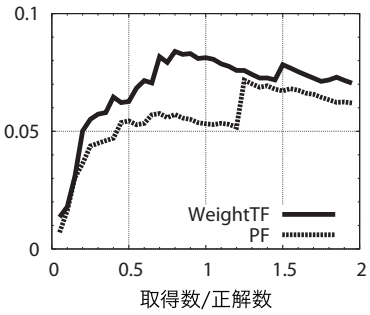


図 6　下位語抽出

表 6　抽出例（“magician” の同義語）

得られた語	PF	得られた語	WeightTF
*wizard	49	*sorcerer	9.78
*illusionist	47	clown	7.37
*sorcerer	43	*wizard	5.93
magic	40	*illusionist	5.62
clown	37	*shaman	5.55
*juggler	36	*witch	5.09
priest	36	*enchanter	4.30
*artist	35	priest	4.07
*witch	35	*juggler	4.01
warrior	34	*artist	3.59

より取得した同義語上位 10 件を表 6 に示す．ここで，‘∗’ がついているものは，Roget’s Thesaurus に登録されているものである．

5　まとめ

本論文では，検索エンジンのスニペットを用いることで，ウェブをコーパスとして利用する関連語抽出の手法を提案した．本手法を用い，ある関係にある語ペアの正例となるデータを用意すれば，対象語に対して，指定した関係にある語の順位付けされたリストを自動的に得ることができる．ウェブをコーパスとして用いているため，新しい語や意味の変化にも対応できることが利点である．提案手法は，係り受け解析を行っている Lin の手法とほぼ同等の精度であり，タグ付けや構文解析を行っていない点で意義のある手法といえる．

今後の課題として，さらなる精度の改善が挙げられる．実験結果を見ると，多くの場合，候補語は取れており，ステップ 3 のランキングを改良することで，精度の改善が可能であると考えられる．例えば，スニペットだけではなく検索にヒットしたページも利用してパターンを抽出する，様々な特徴を用いてランキングに対して機械学習を行うなどの工夫が考えられる．また，表 6 を見ると，‘∗’ がついていない（Roget’s Thesaurus に登録されていない）語であっても，“clown” などは同義語としてふさわしいと言える．このような語は多く存在すると考えられるので，今後は既存のコーパスによる評価だけでなく，コストはかかるが，人手による評価を行っていく必要がある．特定の関係性を，パターンを用いて精度よく抽出する手法は，ウェブからオントロジーや知識を抽出する研究で基盤となるものであり，本研究の方向性をさらに進め，さまざまな関係性に適用可能な手法を構築していきたいと考えている．

参考文献

1) D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International World Wide Web Conference (WWW-07)*, pages 757-766, 2007.

2) D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 19th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 768-774, 1998.

3) G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4): pages 235-244, 1990.

4) P. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML-01)*, pages 491-502, 2001.

5) R. Girju, A. Badulescu, and D. Moldovan. Automatic Discovery of Part-Whole Relations. In

Computational Linguistics, 32(1), pages 83-135, 2006.

6) M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539-545, 1992.

7) M. Pasca. Organizing and searching the World Wide Web of facts - step two: harnessing the wisdom of the crowds. In *Proceedings of the 16th International Conference on World Wide Web (WWW-07)*, pages 101-110, 2007.

8) M. Porter. An Algorithm for Suffix Stripping. *Program*, 14, pages 130-137, 1980. Accesible at http://www.tartarus.org/ martin/PorterStemmer/.

9) P. Cimiano and J. Wenderoth. Automatic acquisition of ranked qualia structures from the web. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 888-895, 2007.

10) P. Pantel, D. Ravichandran, and E. Hovy. Towards Terascale Knowledge Acquisition. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 771-777, 2004.

11) P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 113-120, 2006.

12) P. Roget. *Thesaurus of English words and phrases*. Longmans, Green and Co., 1911.

13) R. Snow, D. Jurafsky, and A. Ng. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*, 2004.

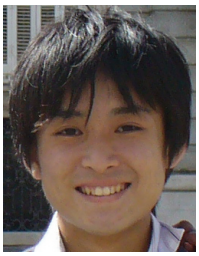
14) 榊剛史, 松尾豊, 内山幸樹, 石塚満. Web 上の情報を用いた関連語のシソーラス構築について. 自然言語処理, Vol. 14, Number 2, pages 3-31, 2007.

15) Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida and M. Ishizuka. POLYPHONET: an advanced social network extraction system from the web. In *Proceedings of the 15th International Conference on World Wide Web, (WWW-06)*, 2006.

問合せ先

〒 113-8656
東京都文京区本郷 7-3-1
東京大学大学院情報理工学系研究科
ボレガラ　ダヌシカ
Tel: 03-5841-6751
Fax: 03-5841-6070
E-mail: danushka@iba.t.u-tokyo.ac.jp

著者紹介



渡部 啓吾 (非会員) 2008 年東京大学工学部電子情報工学科卒業. 2010 年同大学院情報理工学系研究科修士課程修了. 現在:DeNA. ウェブマイニング, 自然言語処理に興味を持つ.

Bollegala Danushka (非会員) 2005 年東京大学工学部電子情報工学科卒業. 2007 年同大学院情報理工学系研究科修士課程修了. 2009 年同研究科博士課程修了. 博士 (情報理工学). 現在：同研究科・助教. 自然言語処理, 機械学習, ウェブマイニングに興味を持つ. WWW,ACL などの会議を中心に研究成果を発表.

松尾 豊 (非会員) 1997 年東京大学工学部電子情報工学科卒業. 2002 年同大学院博士課程修了. 博士 (工学). 同年より, 産業技術総合研究所情報技術研究部門勤務, 2005 年 10 月よりスタンフォード大学客員研究員. 現在：東京大学工学系研究科総合研究機構・准教授. 人工知能, 特に高次ウェブマイニングに興味を持つ. 人工知能学会, 情報処理学会, AAAI の各会員.

石塚 満 (非会員) 1971 年東京大学工学部電子卒，1976 年同大学院工学系研究科博士課程修了．工学博士．同年 NTT 入社，横須賀研究所勤務．1978 年東京大学生産技術研究所・助教授，(1980-81 年 Purdue 大学客員准教授)，1992 年東京大学工学部電子情報工学科・教授．現在：同大学院情報理工学系研究科・教授．研究分野は人工知能,Web インテリジェンス, 意味計算, 生命的エージェントによるマルチモーダルメディア.IEEE,AAAI, 人工知能学会（元会長）, 電子情報通信学会, 情報処理学会等の会員.