原著論文 Original Paper

関係の対称性および予測語を用いた関係検索の性能向上法

Improving Relational Search Performance using Relational Symmetries and Predictors

後藤 友和 Tomokazu Goto
東京大学大学院 情報理工学研究科 Graduate School of Information and Technology, The University of Tokyo
goto@mi.ci.i.u-tokyo.ac.jp

グエン トアンドゥク Nguyen Tuan Duc
（同　上）
duc@mi.ci.i.u-tokyo.ac.jp

ボレガラ ダヌシカ Danushka Bollegala
（同　上）
danushka@iba.t.u-tokyo.ac.jp,http://www.iba.t.u-tokyo.ac.jp/~danushka/

石塚 満 Mitsuru Ishizuka
（同　上）
ishizuka@i.u-tokyo.ac.jp, http://www.miv.t.u-tokyo.ac.jp/ishizuka/

keywords: relational similarity, relational search, proportional analogy, SAT

Summary

Relational similarity can be defined as the similarity between two semantic relations R and R' that exist respectively in two word pairs (A,B) and (C,D). Relational search, a novel search paradigm that is based on the relational similarity between word pairs, attempts to find a word D for the slot ? in the query {(A,B), (C,?)} such that the relational similarity between the two word pairs (A, B) and (C, D) is a maximum. However, one problem frequently encountered by a Web-based relational search engine is that the inherent noise in Web text leads to incorrect measurement of relational similarity. To overcome this problem, we propose a method for verifying a relational search result that exploits the symmetric properties in proportional analogies. To verify a candidate result D for a query {(A, B), (C, ?)}, we replace the original question mark by D to create a new query {(A,B),(?,D)} and verify that we can retrieve C as a candidate for the new query. The score of C in the new query can be seen as a complementary score of D because it reflects the reliability of D in the original query. Moreover, transformations of words in proportional analogies lead to relational symmetries that can be utilized to accurately measure the relational similarity between two semantic relations. For example, if the two word pairs (A,B) and (C, D) show a high degree of relational similarity then the two word pairs (B,A) and (D,C) also have a high degree of relational similarity. We apply this idea in relational search by using symmetric queries such as {(B, A), (D,?)} to create six queries for verifying a candidate answer D to improve the reliability of the verification process. Our experimental results on the Scholastic Aptitude Test (SAT) analogy benchmark show that the proposed method improves the accuracy of a relational search engine by a wide margin.

1. ま え が き

二つの概念 A と B の属性間の対応 (correspondence) が別の二つの概念 C と D の属性間の対応と類似している場合（A,B) 対と（C,D) 対は関係類似性を持っていると定義される．本論文では単語を用いて概念を表し，扱う関係の種類（例：hypernymy, meronymy, synonymy など）については限定しない．例えば，属性類似性が高い単語の組み合わせとして，cat と lion が挙げられる．それぞれ「鋭い歯を持つ」,「4 本の足で歩く」などの共通した属性を持つためである．一方，関係類似性が高い単語対の組み合わせとしては,（bird，ostrich）,（cat，lion）という単語対が考えられる．それぞれの単語対が，前者の単語を〜とし，後者の単語を…としたとき,「〜は大きな…」という関係を共通して持つためである．関係類似性を定義する際には単語ではなく単語対を必要とすることに注意されたい．また，bird に対する ostrich は cat に対する lion と等しいことを bird:ostrich::cat:lion と書き，これを比例的類推 (proportional analogy) であると言う [Turney 06b]．つまり，4 つの単語 A,B,C,D からなる 2 つの単語対 (A,B), (C,D) における関係類似度が高いときに A:B::C:D が成り立つ．

(bird, ostrich), (cat, lion) のように 4 つの単語全てが与えられている状態で，関係を成す単語を 1 つ消し，それ

を「?」に置き換えて {(bird, ostrich), (cat, ?)} という単語対の組を作成する．このとき,「?」に当てはまる語として関係類似度が高くなる語を選ぶことで lion を導ける．このように，2 つの単語対において「?」に相当する単語を見つけるタスクを関係検索という [Kato 09, Duc 10]．関係検索が実現できると，例えば {(Japan, Mt. Fuji), (U.S, ?)} というクエリを用いることでアメリカにおける富士山的存在を取得することができる．富士山は日本に対して「日本で一番高い」という関係以外にも「日本で一番綺麗な山の」,「日本で湖を持つ山の」,「日本の成層型火山である」などといった多数の関係を持っており，従来のキーワードマッチングベースの検索エンジンではこれらの関係を全て列挙して，アメリカにおける富士山的存在を探すことは難しい．このように，求める語が他の語と多数の関係を持つ場合に関係検索は有効である．

以下，関係検索に用いるクエリ Q_{RS} を，3 つの単語 A, B, C を用いて Q_{RS} = {(A, B), (C, ?)} と表す．(A, B) は A, B による単語対を表し，{(A, B), (C, ?)} は入力ペアが (A, B) で，質問ペアが (C, ?) である関係検索のクエリを表す．質問ペアでは，関係検索の答えが入る位置が「?」で表され，これはプレースホルダとも呼ばれる．また，最初のクエリから得られた D を用い，クエリ中の他の単語をプレースホルダにしたときに取得が期待できる語のことを予測語という．例えば，Q_{RS} = {(A, B), (C, ?)} から得た D を用いたクエリ Q_{RS} = {(A, B),(?, D)} においては予測語 C を得ることが期待できる．

関係検索は主に Web のデータを用いて実現されているため [Kato 09, Duc 10]，ノイズが混じりやすく，それによる性能低下が起こるという問題がある．本論文では，関係検索における予測語の期待性と，関係の対称性による複数の関係検索のクエリを用いることで検索結果の検証を行い，関係検索の性能を向上させる手法を提案する．

2. 関係の対称性

“John is Tom’s father”という文と “Tom is John’s child” という文があるとする．後者の文は前者の文中の John と Tom の位置を入れ替え，語間の関係を father から child に変更したものである．このとき，両者の文は共に John と Tom の親子関係を言っており，その意味は変わらない．このように，文中の 2 単語 (X, Y) の出現箇所を入れ替えたときに元々の関係を保持するような関係 R’ を持つことを対称的な関係であると定義する．なお，本論文における手法では対称的な関係のみを扱う．対称的な関係は，元々の文が持っていた関係を R とすると，

$$\mathrm{R}(X,Y) = \mathrm{R}'(Y,X) \tag{1}$$

と書ける．ここで，R(X, Y) は X と Y が関係 R で結合して，ある意味を形成していることを表す．つまり，式 (1) は X と Y の位置を入れ替えた際に，入れ替える前の元の意味を保持する関係 R’ が存在し，入れ替え後も元々の文と意味が変わらないことを表す．

一方，関係類似性においても対称性を定義することができる．4 つの単語 A，B，C，D が A:B::C:D であるとき，表 1 で示す 4 通りの，共通の関係 R および対称的な関係 R’ を持つ対称的な組み合わせを定義できる．R と対称的な関係にある R’ が存在し，R(A, B) = R’(B, A) かつ R(C, D) = R’(D, C) であるならば，A:B::C:D より，R(A, B) = R(C, D) が成り立つため，R’(B, A) = R’(D, C) となり，A:B::C:D は B:A::D:C と等しい (表 1，番号 2)．また，A:B::C:D が関係 R で結ばれているとき，それぞれの単語対 (A, B) と (C, D) の順序を変えても関係 R は変わらない．このことから，A:B::C:D は C:D::A:B と等しい (表 1，番号 3)．さらに，表 1 の番号 2 と表 1 の番号 3 から C:D::A:B と D:C::B:A が等しいことが導ける (表 1，番号 4)．他にも A:C::B:D などの組み合わせが考えられるが，このとき例えば A = ostrich, B = bird, C = lion, D = cat とすると，ostrich:lion::bird:cat となり,「〜は大きな…」という A:B::C:D が持っていた共通の関係 R を失うため，A:B::C:D と A:C::B:D は等しいとは言えない．他に考えられる A，B，C，D の組み合わせに対しても同様であるため，表 1 に示す 4 通りの関係のみが A:B::C:D と等価である．

表 1 関係類似性に基づく対称的な比例的類推の組み合わせ

番号	組み合わせ
1	A:B::C:D
2	B:A::D:C
3	C:D::A:B
4	D:C::B:A

3. 関係検索

関係検索において Q_{RS} = {(A, B), (C, ?)} から D を得るまでには，

(1) A, B 間の関係 R の抽出
(2) R を用いて，C と R の関係にある単語 D を抽出する
(3) D をある尺度に基づいてランキングし，ユーザに提示する

という 3 つのステップを必要とする．3 章では，本研究におけるこれら 3 ステップの詳細および，予測語および対称性を用いた D の検証方法について述べる．

3·1 A, B 間の関係抽出

A と B の間の関係を取得するに際し，本研究では検索エンジンのスニペットを用いた．スニペットとは，図 1 で表されるもので，検索条件に合致したページにおいて検索語が含まれる箇所を表す抜き書きのことである．スニペットにはクエリで使われた文字列が含まれ，かつその周囲のみが抽出されて提示されている．文章情報を取得する際には文書単位で取得されることが多いが，関係

... the Chinese mythology, big **cat such as lion** is considered ...

図 1 “cat * * * lion” というクエリに対するスニペットの一部

検索ではクエリで使用した単語同士の関係さえ取得できれば良いので，本研究ではスニペットを用いる．

多くの検索エンジンではクエリをダブルクォーテーション「“」と「”」で囲うことにより，クエリの語順を保持したまま検索を行うことができる．また，ワイルドカード「*」も多くの検索エンジンでサポートされており，*は 1 つ以下の単語にマッチさせるのに使える．例えば,「“ostrich * * * bird”」というクエリを用いたとき，ostrich と bird に囲まれる最大 3 単語が*と置き換わる形でスニペットに出力される．このことを用いて，A, B 間の関係の抽出を行う．ここでは関係は全てパターンと呼ばれる文字列で表される．パターン文字列は，A, B を含む文字列において A，B をそれぞれ変数 X，Y に置き換えたものである．例えば，ostrich と bird から “ostrich is a large bird” という文字列を取得できたとき，そのパターン文字列は ostrich と bird をそれぞれ変数 X，Y に置き換えた，X is a large Y となる．ただし，先に現れたものを X，後に現れたものを Y とする．

本研究では関係を表すパターン文字列を含むスニペットを取得するために検索エンジンに対して A と B の間に n 個の*を配置してクエリを作成する．なお，本研究では n = 3 とした．つまり，クエリは「“A * * * B”」となる．このクエリを用いたときに検索エンジンから得られるスニペットには A と B との間に最大 n 個の単語が入った文字列が含まれる．次に，スニペットを文単位に区切り，A，B 両方を含む (n+2)-gram 以下で部分文字列を取得し，A と B を X と Y に置き換えたものをパターン文字列として保持する．例えば，図 1 に示したスニペットが取れたとき，cat と lion を含む 5-gram 以下の部分文字列を抽出すると,「**cat** such as **lion**」「big **cat** such as **lion**」「**cat** such as **lion** is」という 3 つの部分文字列を取得でき，cat と lion をそれぞれ X と Y に置き換えることで次の 3 つのパターン文字列,「X such as Y」「big X such as Y」「X such as Y is」が得られる．こうして得られたパターン文字列を関係 R として，C と R の関係にある単語 D を抽出するのに用いる．

3·2 D の抽出

3·1 節で取得したパターン文字列の集合を $\mathfrak{S}$ とする．Q_{RS} = {(A, B), (C, ?)} のとき，S $\in \mathfrak{S}$ となる各パターン文字列 S に対して，S に含まれる X を C に，Y を*に置き換える．このとき，X，Y はそれぞれ A，B が置き換わったものであり，A → X → C および B → Y → *という，A から C および B から*への対応付けを保持する必要があることに注意されたい (3·3 節参照のこと)．このように，S に含まれる X を C に，Y を*に置き換えることを S(C, *) と書くことにする．3·1 節では，関係を*を用いて抽出していたが，今回は単語を抽出するため，?の部分を*に置き換えている．そして，検索エンジンに対してクエリ「“S(C, *)”」を実行してスニペットを取得する．得られたスニペットから，*が置き換わった部分の単語を抽出し，D とする．

こうして得られた各 D を出力する際には D の集合 $\mathfrak{D}$ 内で (A, B) と (C, D) 間の関係類似度が高い順に並ぶのが望ましい．そこで次に，D $\in \mathfrak{D}$ となる各 D に対してスコア付けを行う．

3·3 D のスコア付け

Q_{RS} = {(A, B), (C, ?)} における D の候補の妥当性を表すスコアを score(D) と表す．本研究では score(D) を式 (2) で定義する．

$$
\begin{aligned}
&\mathrm{score(D)} \\
&= \sum_{\mathrm{S}\in\mathfrak{S}_\mathrm{D}} P((\mathrm{C},\mathrm{D})|\mathrm{S})P(\mathrm{S}|(\mathrm{A},\mathrm{B})) \\
&= \sum_{\mathrm{S}\in\mathfrak{S}_\mathrm{D}} \frac{P((\mathrm{C},\mathrm{D}),\mathrm{S})}{P(\mathrm{S})} \cdot \frac{P((\mathrm{A},\mathrm{B}),\mathrm{S})}{P((\mathrm{A},\mathrm{B}))} \\
&= \sum_{\mathrm{S}\in\mathfrak{S}_\mathrm{D}} \frac{\frac{\mathrm{h}(\text{“S(C,D)”})}{\mathrm{N}}}{\frac{\mathrm{h}(\text{“S(*,*)”})}{\mathrm{N}}} \cdot \frac{\frac{\mathrm{h}(\text{“S(A,B)”})}{\mathrm{N}}}{\frac{\mathrm{h}(\text{“A***B”})}{\mathrm{N}}} \\
&= \sum_{\mathrm{S}\in\mathfrak{S}_\mathrm{D}} \frac{\mathrm{h}(\text{“S(C,D)”})}{\mathrm{h}(\text{“S(*,*)”})} \cdot \frac{\mathrm{h}(\text{“S(A,B)”})}{\mathrm{h}(\text{“A***B”})}
\end{aligned}
\tag{2}
$$

ここで，$\mathfrak{S}_\mathrm{D}$ は D を取得した際に用いたパターン文字列 S の集合を，P は文書における文字列の出現確率を，h は文字列を検索エンジンに投げたときの検索ヒット件数を表す．また，式 (2) において (A, B) は A と B が特定の順番かつ n 個離れた箇所に出現することを意味しており，一般的に $\mathfrak{S}_\mathrm{D}$ は非対称な関係を表すパターン (例：「X such as Y」 と 「Y such as X」) も含むため 順不同ではないことに注意されたい．ここで，確率を表すのに必要な母集合はウェブに存在する全ての文書数 N となるが，N は式 (2) 中の全ての確率の分母として現れ，それぞれを N で割ることで消えるため，計算をする上で考慮する必要はない．

D を取得する際に用いるパターン文字列は，(A, B) を表す際によく用いられるパターン文字列であることが望ましい．どれが (A, B) で多く出現しやすいパターン文字列であるかは $P(\mathrm{S}|(\mathrm{A},\mathrm{B}))$ で表すことができる．さらに，(A, B) から取得したパターン文字列を用いて出現しやすい (C, D) を求めるには，$P((\mathrm{C},\mathrm{D})|\mathrm{S})$ を求めればよい．つまり，$P((\mathrm{C},\mathrm{D})|\mathrm{S})P(\mathrm{S}|(\mathrm{A},\mathrm{B}))$ は，あるパターン文字列 S が (A, B) からどれだけ出現しやすいか，また，その S を使ってどれだけ (C, D) が出現しやすいかを表している．これは，(C, D) がパターン文字列 S を通してどれだけ (A, B) と近いかを現しており，一種の関係類似度を測っているといえる．また，最終的な式の形を見ると，

表2 対称的な関係検索のクエリ

名前	対称的なクエリ Q'_{RS}	予測語	スコア
sym1	{(A, B),(?, D)}	C	$s_1 = \mathrm{score(C)}$
sym2	{(B, A),(D, ?)}	C	$s_2 = \mathrm{score(C)}$
sym3	{(C, D),(A, ?)}	B	$s_3 = \mathrm{score(B)}$
sym4	{(C, D),(?, B)}	A	$s_4 = \mathrm{score(A)}$
sym5	{(D, C),(B, ?)}	A	$s_5 = \mathrm{score(A)}$
sym6	{(D, C),(?, A)}	B	$s_6 = \mathrm{score(B)}$
sym7	{(A, B),(C, ?)}	D	$s_7 = \mathrm{score(D)}$
sym8	{(B, A),(?, C)}	D	$s_8 = \mathrm{score(D)}$

検索エンジンのヒット件数のみを用いており，他のものに依存していないことが分かる．なお，ウェブ検索エンジンのヒット件数はその近似値であることが多いが，そうしたヒット件数を用いて単語間の関係が計測できる手法 [Bollegala 07] が提案されており，それにならって本研究においてもヒット件数を用いている．

3·4 予測語と対称的なクエリによるノイズの影響の削減

関係検索のクエリ Q_{RS} = {(A, B), (C, ?)} において検索結果の単語を D とすると，クエリに含まれる単語対と共通の関係 R を用いて R(A, B)，R(C, D) と書くことができる．ここで，関係検索の定義より，D は A:B::C:D が成り立つような単語であるため，表 1 で定義した 4 つの対称性も成り立つ．Q_{RS} = {(A, B), (C, ?)} から取得できた D を実際にプレースホルダの位置に当てはめると，Q_{RS} = {(A, B), (C, D)} となる．ここから，表 1 による 4 つの対称性を用い，質問ペアのうち 1 つをプレースホルダに置き換えることで，対称的な関係検索のクエリ Q'_{RS} を定義することができる．Q'_{RS} は表 2 に示す sym1 から sym8 までの 8 種類考えることができる．ここでは，D を取得する際にも対称的なクエリを考えることができるため，それを便宜上 sym7，sym8 とした．そして，各クエリを用いたときに取得できる予測語のスコアをそれぞれ s_1 から s_8 とした．

D を Q_{RS} = {(A, B), (C, ?)} から得た語だとするとき，もし D が A:B::C:D を満たす単語であるならば，Q_{RS} = {(A, B),(?, D)} からは予測語 C を得ることが期待できる．逆に，D が A:B::C:D を満たさない単語であるならば Q_{RS} = {(A, B),(?, D)} からは予測語 C を得られない．このことを用いることにより，関係検索の出力 D の妥当性をチェックすることができ，ウェブのノイズの影響を削減することができる．このような予測語は表 2 に示す sym1 から sym6 までの 6 通りあり，D を取得するために用いる sym7 と sym8 を加え，計 8 通りのクエリを用いて D の妥当性を評価できる．これら 8 通りのクエリを用いて得られる D の最終的なスコア FinalScore(D) を次のように定義する．

$$\mathrm{FinalScore(D)} = \sum_{i=1}^{8} u_i s_i \tag{3}$$

ここで，u_i は，i 番目のスコアを用いるか否かを表す値であり 0 か 1 の値を取る．本論文では 4 章で実験的に $u_1 \ldots u_8$ の組み合わせを求めている．D が A:B::C:D を満たさない場合，そもそも sym1 から sym6 における予測語が表れないか，予測語が出現したとしてもそのスコアが低くなると考えられるので，式 (3) のスコアが高ければ高いほど，より A:B::C:D を満たす D であるといえる．

3·5 関係検索の一連の流れ

図 2 は関係検索の実行における一連の流れの例を示している．まず，Q_{RS} = {(ostrich, bird), (lion, ?)} という関係検索のクエリに対し，前者の単語対から “ostrich * * * bird” というクエリを作成し，検索エンジンに投げる．そして，検索エンジンの出力から “the **ostrich is the largest bird** in size and weight on earch ...”というスニペットが得られたとき，そこから ostrich と bird を含む 5-gram 以下の文字列である “ostrich is the largest bird”を抽出する．この文中における ostrich と bird をそれぞれ X と Y に置き換え，“X is the largest Y”というパターン文字列 S を得る．そして，パターンに質問ペア (lion, ?) を代入することで S(lion, *) = “lion is the largest *” というクエリを作成し，再び検索エンジンに投げる．再び検索エンジンの出力から “the **lion is the largest cat**, while some...” というスニペットが得られたとき，そこから*が置き換わった語を抽出することで候補語の 1 つである cat を得ることができる．上記の例では lion，cat と共に現れたパターンは “X is the largest Y” であり，これが式 (2) の $\mathfrak{S}_D$ に相当する．なお，例には登場していないが，他にも “X is the big Y”などもパターンとして出現することがある．このとき，式 (2) では，$\mathfrak{S}_{cat}$ = {“X is the largest Y”,“X is the big Y”} となる．

そして，得られた D = cat に対し，式 (2) を用いてスコア付けを行う．$\mathfrak{S}_{cat}$ = {“X is the largest Y”,“X is the big Y”} なので，

$$\begin{aligned}&\mathrm{score(cat)} = \\ &\frac{\mathrm{h(“lion\ is\ the\ largest\ cat”)}}{\mathrm{h(“*\ is\ the\ largest\ *”)}} \cdot \frac{\mathrm{h(“ostrich\ is\ the\ largest\ bird”)}}{\mathrm{h(“ostrich\ *\ *\ *\ bird”)}} + \\ &\frac{\mathrm{h(“lion\ is\ the\ big\ cat”)}}{\mathrm{h(“*\ is\ the\ big\ *”)}} \cdot \frac{\mathrm{h(“ostrich\ is\ the\ big\ bird”)}}{\mathrm{h(“ostrich\ *\ *\ *\ bird”)}}\end{aligned}$$

となる．これを $Q_{RS} = \{(\mathrm{bird}, \mathrm{ostrich}), (?, \mathrm{lion})\}$ についても行い，cat を得たとする．

最後に，対称性を用いて D の妥当性を評価する．D = cat であったから，そこから sym1 = {(ostrich, bird), (?, cat)}, sym2 = {(bird, ostrich), (cat, ?)}，sym3 = {(lion, cat), (ostrich, ?)}，sym4 = {(lion, cat), (?, bird)}，sym5 = {(cat, lion), (bird, ?)}，sym6 = {(cat, lion), (?, ostrich)} という 6 種類の対称的なクエリを用いて得られたそれぞれの予測語のスコアと，cat に関する 2 種類のスコア (sym7, sym8) を用いて FinalScore(cat) を算出する．これを cat 以外に取得できた他の単語についても行い，D のランキングを行うことでユーザに結果を返す．

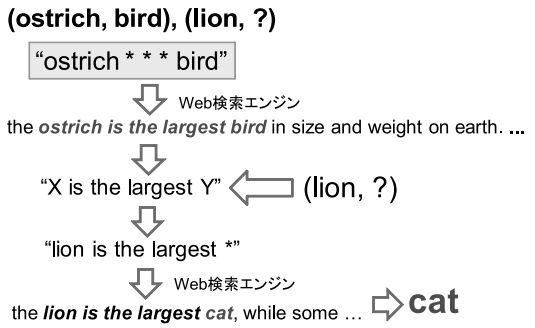

図 2 $\mathrm{Q_{RS}} = \{(\mathrm{ostrich}, \mathrm{bird}), (\mathrm{lion}, ?)\}$ に対する関係検索の実行例

表 3 SAT のアナロジー問題の例

Stem pair	ostrich	bird
1	lion	cat
2	goose	flock
3	ewe	sheep
4	cub	bear
5	primate	monkey

4. 評 価 実 験

関係類似性測定手法を評価するためのベンチマークとして Scholastic Aptitude Test (SAT) データセットが先行研究では使われている．本論文で提案する関係の対称性に着目した関係検索の手法を，SAT 問題を解くために応用することによって，単語対間の関係を調べる先行研究と容易に比較でき，その有効性を客観的に評価できる．SAT データセットに含まれる問題の例を表 3 に示す．SAT のアナロジー問題は Stem pair と呼ばれる問題の単語対と，回答の選択肢となる 5 つの候補の単語対から成り，データセット内には同様の問題が全部で 374 問存在する．回答者は問題単語対の関係に最も類似する単語対を候補の単語対の集合から 1 つ選び，回答する．表 3 の場合，ostrich と bird の関係に最も近くなるような候補の単語対は lion と cat であるため，正解は 1.(lion,cat) となる．なぜなら，この 2 つの単語対は「X is a large Y」という関係を共通して持っているからである．既存の関係類似度測定の研究では問題の単語対と最も関係類似度が高くなるような候補の単語対を選ぶことで正解を得てきた．

関係検索を用いる場合では，$\mathrm{Q_{RS}} = \{(\mathrm{A}, \mathrm{B}), (\mathrm{C}, ?)\}$ というクエリの単語対をそれぞれ Stem pair と候補ペアに含まれる単語に当てはめることで SAT の問題を解くことができる．具体的には，A と B をそれぞれ Stem pair に含まれる単語に置き換え，C を候補ペアの最初の単語に割り当てれば良い．こうして $\mathrm{Q_{RS}}$ を作成し，D を検索する．そして，「?」に当てはまった，候補ペアに含まれる単語のうち C でない単語（ラストワード）のスコアを見る．これを他の全ての候補に対しても行い，それぞれのラストワードのスコアを比較し，最もスコアが高いものを選ぶことによって SAT への回答とする．例えば，表 3 の場合，A = ostrich, B = bird, C = lion とし，D として出てくるラストワード (= cat) のスコアを見る．これを他の 4 つの候補に対しても行い，取得できたラストワードのスコアを比較する．その結果，最も高いスコアを出した候補を SAT の答えとして出力する．システムの出力した SAT の答えと SAT データセットに用意された正解と照合することでシステムによる結果が正しいかどうかを判断できる．本研究ではこの正解率を見ることで，精度，再現率，F 値を求め，対称性を用いる前後においての性能比較および先行研究との比較に用いた．

表 4 SAT データセットにおける正解率

番号	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	精度	再現率	F 値	順位
1	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**.511**	**.447**	**.476**	**1**
2	0	0	0	0	1	0	1	1	.509	.439	.471	2
3	0	0	0	0	1	1	0	1	.506	.433	.467	3
4	0	0	1	1	1	1	1	1	.491	.444	.466	4
5	0	0	1	1	1	0	1	1	.488	.441	.463	5
6	0	0	1	0	1	1	1	1	.488	.439	.462	6
7	0	0	1	0	1	1	1	0	.494	.433	.462	7
8	0	0	1	1	1	1	0	1	.487	.439	.461	8
9	0	0	0	0	1	0	0	1	.511	.420	.461	9
10	0	0	0	0	1	0	1	0	.511	.420	.461	10
11	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**.442**	**.404**	**.422**	**56**
12	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**.509**	**.382**	**.437**	**33**
13	0	0	0	0	0	1	0	0	.586	.345	.434	37
14	0	0	1	0	0	0	0	0	.421	.313	.359	234
15	0	0	0	0	0	0	0	1	.420	.305	.353	239
16	1	0	0	0	0	0	0	0	.383	.307	.341	246
17	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**.490**	**.257**	**.337**	**250**
18	0	1	0	0	0	0	0	0	.336	.262	.294	253
19	0	0	0	1	0	0	0	0	.352	.246	.290	255

なお，SAT アナロジー問題を解くための関係検索は検索エンジンとして Yahoo! Boss API[*1]を用い，言語は英語を対象とした．また，パターン文字列集合 $\mathfrak{S}$ を取得するための “A * * * B”というクエリに対しては 1000 件のスニペットを取得し，D を取得するための各パターン文字列 $S \in \mathfrak{S}$ に対する S(C, *) というクエリに対しては各 50 件ずつスニペットを取得した．

実験結果を表 4 に示す．番号 1 から 10 は F 値に関して上位 10 個となる対称性の組み合わせを，番号 11 は全ての対称性を用いた組み合わせを，そして番号 12 から 19 は，それぞれの対称性を 1 つだけ用いた対称性の組み合わせを表す．対称性の組み合わせ $(u_1, ..., u_8)$ は，式 (3) における u_i に対して，i 番目のスコア (s_i) を使用する時に $1(u_i = 1)$ を割り当て，使用しない時に $0(u_i = 0)$ を割り当てたものである．例えば番号 1 では，u_1 = 0, u_2 = 0, u_3 = 0, u_4 = 0, u_5 = 1, u_6 = 1, u_7 = 1, u_8 = 1 となり，s_1 から s_4 の対称性は全く使用せず，s_5 から s_8 の対称性は全て使用することを表す．これら u_i の組み合わせは $2^8 = 256$ 通りから，全てが 0 の場合を除いた 255 通り存在する．よって F 値の順位は 1 から 255 まで存在する．また，表 4 に出てくる精度 (precision)，再現率 (recall)，

*1 http://developer.yahoo.com/search/boss/

F 値 (F-measure) はそれぞれ先行研究に従って，

$$\text{precision} = \frac{\text{正解数}}{\text{回答できた問題数}}, \tag{4}$$

$$\text{recall} = \frac{\text{正解数}}{\text{全問題数}}, \tag{5}$$

$$\text{F} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

と定義する．ここで，精度を表す式にある「回答できた問題数」は，Q_{RS} = {(A, B), (C, ?)} または Q_{RS} = {(B, A), (?, C)} において，候補ペア 5 つのうち 1 つでも D としてラストワードを抽出できた問題の数を表す．なお，1 位との統計的有意性を検証するために McNemar テストを行った結果，2 位から 9 位までにランクされた手法は統計的に見て，有意水準を 0.05 としたとき，1 位にランクされた手法と有意な差が無いことが明らかになった．

5. 考　　察

本研究では，D に関する対称性を使う前と対称性を使った後での F 値の比較，および SAT データセットを用いた先行研究との性能比較によって評価を行う．D に関する対称性を使わない場合は表 4 において番号 17 で表される．これは，Q_{RS} = {(A, B), (C, ?)} から得られた D のスコアに等しく，本手法において対称性を使用する前後における比較対象となる．つまり，番号 17 との差が大きければ大きいほど本手法による対称性を用いた D の検証が有効であると言える．

5·1 表 4 における考察

表 4 を見ると，番号 1 のときに最も F 値が高くなっている．これは，単に全ての対称性を使った場合 (番号 11) よりも良い結果になっている．また，単独で対称的なクエリを用いた場合では，番号 12，つまり sym6 = Q_{RS} = {(D, C), (?, A)} を用いたときが最も F 値が高い．また，特筆すべき点として，対称性を使った D の検証を行う前 (番号 17) に比べて F 値は 33.7% から 47.6% へと 13.9% 上昇しており，検証を行う前に比べて 141% の F 値になっている．このことから，本手法による対称性を用いた D の検証が関係検索の性能を向上させていることが分かる．なお，番号 1 のときは関係検索を 3 回多く行う必要があるが，それぞれの検索は独立して行うことができるため，並列して実行することで処理時間の増加を抑えることができる．

次に，順位が 10 位までの u_i の組み合わせをみると，表 2 中の s_1，s_2 は一切使われておらず，逆に s_5 や s_6 は多く使われている．さらに，s_5 と s_6 は単独で用いたときの F 値も高い．特に s_5 に関しては上位 10 件全てにおいて使われていることが分かる．このように，s_5 および s_6 による検証が特に有効に働いたのは，誤った D を用いた際に，他と比べて s_5 と s_6 からは予測語を取得しにくいためだと考えられる．A, B を入力ペアに持つ場合，SAT において A と B は関係を持つことが保証されているため，多くの関係が取得できる．そのため，誤った D を用いても予測語を取得できることがある．一方，入力ペアが (D, C) であり，D が誤っている場合，入力ペアにおける D と C の関係は取得できないため，予測語も取得できない．また，D は Q_{RS} = {(A, B), (C, ?)} から取得したので，C と D の順番が保たれるようなクエリにおいて，D が誤っていても少なからず C と D は関係を持つ．しかし，s_5 と s_6 のような C と D の順番が保たれない対称的なクエリでは，D が誤っている場合，C と D の関係は取得できないため，予測語も取得できない．これらの要因が s_5 と s_6 による検証の有効性を高めていると考えられる．

5·2 既存の関係類似度測定研究との比較

表 5 は関係類似性の研究における SAT データセットによる実験結果を提案手法と，先行研究による 27 件の手法を比較したものである [Turney 03, Turney 06b, Bollegala 08, Mangalath 04, Veale 04, Bicici 06, Turney 05, Bollegala 09, Turney 06a]．なお，ここでは過去の研究にならい，F 値を用いた．SAT は 5 問から正解の 1 問を選ぶのだから，ランダムに解けば正解率は 20%となる (Random guessing)．また，実際に SAT を解いた大学受験者たちの平均は 57.0%となっている (Human)．今回の実験で得られた F 値 (PROPOSED) は最大で 47.6%であり，手法全体では 4 番目となるスコアである．

関係類似度測定の研究における最大のスコア 56.1%は下回っているが，本手法は以下の理由から単純に (A, B) と (C, D) 間の関係類似度を測定する場合に比べて精度・再現率が下がる傾向にあると考えられる．表 5 中の関係類似度測定の研究では，本手法とは異なり，A，B，C，D の全てが与えられた状態で SAT を解いており，(A，B) と (C, D) 間の関係類似度を測定して正解を判断するのみで良かった．しかし，関係検索の場合は A, B, C のみが与えられた状態で「?」に置き換わっている D をまず検索して見つけなければならないため，そもそも D が見つけられないことがあり，これが精度を下げる要因となる．さらに，D を見つけられた場合でも，D 以外の関係検索の適切な語が多く見つかった場合，SAT の正解となる D に関する情報があまり得られなくなり，ラストワードのスコアが低くなる要因となる．例えば，Q_{RS} = {(bird, ostrich), (cat, ?)} としたときに，関係検索の出力候補として jaguar，lion，cheetah，siberian tiger などを得ることができる．これらは全て大きな猫であり，関係検索の結果としては適切である．しかし，SAT の問題としてみると，(bird, ostrich) の問いの答えとして適切なのは (cat, lion) というペアのみなのだから，lion だけがラストワードのスコアに関係する．もし，lion 以外の大きな猫に関

表 5 既存の関係類似度測定研究との SAT データセットにおける性能比較

アルゴリズム	スコア	アルゴリズム	スコア
Similarity:dict [Turney 03]	18.0%	Leacock & Chodrow [Turney 06b]	31.3%
Meronym:substance [Turney 03]	20.0%	Hirst & St.-Onge [Turney 06b]	32.1%
Meronym:member [Turney 03]	20.0%	Resnik [Turney 06b]	33.2%
Holonym:substance [Turney 03]	20.0%	PMI-IR [Turney 06b]	35.0%
Holonym:member [Turney 03]	20.0%	Phrase Vectors [Turney 03]	38.2%
Random guessing	20.0%	SVM [Bollegala 08]	40.1%
Synonym [Turney 03]	20.7%	LSA+Predictation [Mangalath 04]	42.0%
Meronym:part [Turney 03]	20.8%	Veale (WordNet) [Veale 04]	43.0%
Hypernym [Turney 03]	22.7%	Bicici & Yuret [Bicici 06]	44.0%
Antonym [Turney 03]	24.0%	VSM [Turney 05]	47.1%
Hyponym [Turney 03]	24.9%	**PROPOSED**	47.6%
Thesaurus Paths [Turney 03]	25.0%	RELSIM [Bollegala 09]	51.1%
Jiang & Conrath [Turney 06b]	27.3%	Pertinence [Turney 06a]	53.5%
Lin [Turney 06b]	27.3%	LRA [Turney 06b]	56.1%
Similarity:wordsmyth [Turney 03]	29.4%	**Human**	57.0%

する情報が関係検索の結果として多く得られた場合，関係検索 の結果としては良いのに，ラストワードのスコアは悪くなり，再現率を下げる要因となる．

SAT のスコアにおいて本手法を上回った先行研究として RELSIM，Pertinence，LRA がある．RELSIM は SAT データセットを用いた教師有り学習を行っており，学習データを必要としない提案手法に比べて性能は良いが，人手で学習データを作成する必要があり，新しい関係には対応できない．Pertinence と LRA は類義語のシソーラスを用いて回答候補の類義語に関してもパターンを抽出し，それらも用いている．一方，提案手法は類義語シソーラスを用いないため，この付加情報は用いていない．なお，これらの手法は何らかの次元圧縮を行っており（RELSIM はパターンクラスタリング，LRA と Pertinence は特異値分解），同一関係を表す異なる語彙パターンをまとめた上で関係類似度計測を行い，SAT の問題を解くタスクにおいて次元圧縮手法の有効性を示している．これらの手法と比較すると，本手法は類義語のシソーラスや，次元圧縮，機械学習を用いていないが，そのような手法の中では最も高いスコアを出しているといえる．

なお，本論文では関係の対称性に着目し，関係類似度計測に関してその有効性を明らかにするため，次元圧縮手法との組み合わせをあえて考慮しなかったが，今後の研究でその可能性を試みたいと考えている．

6. 関 連 研 究

Turney ら [Turney 05] は，関係類似度を求めるに当たって，ベクトル空間モデル（Vector Space Model：VSM）を用いた．Turney らは，ある単語対 (X, Y) が出現する語彙パターンの数を数えることでベクトルを作成した．語彙パターンは，“X of Y”や “X to Y”といった 128 個の手動で作成された文字列が用いられた．そして，その語彙パターンを検索エンジンに投げ，そこから得られる検索ヒット件数を単語対 (X, Y) からなる単語対ベクトルの各要素値とした．最後に，単語対ベクトル同士のコサイン類似度を測ることにより，VSM における関係類似度の測定を実現した．さらに，Turney [Turney 06b] は，VSM を拡張する形で Latent Relational Analysis(LRA) を提案した．VSM と異なるのは，1.) 語彙パターンは VSM のように手動で作成するのではなく，コーパスから自動で取得されること， 2.) 特異値分解を用いて頻度データのスムージングを行っていること，3.) シソーラスを用いて単語対の類義語もベクトルに含めていること，の 3 つが挙げられる．既存の SAT データセットを用いた関係類似度測定研究の評価では最も良い結果を出しているが，シソーラスを用いた単語対の拡張を行っているため，拡張性が低い．Bollegala ら [Bollegala 09] は VSM におけるベクトルの次元圧縮をクラスタリングを行うことで実現した．さらに，Bollegala らは単語対の近さをマハラノビス距離を用いて表したが，その際に SAT データセットを用いたマハラノビス距離の学習を行うことで精度を向上させている．

Davidov ら [Davidov 07] はある共通した単語クラスを持つ単語群をシードとして与えることで，その単語クラスと関わりのある単語クラスとその間の関係を抽出する手法を提案した．例えば共通した単語クラスが国クラスであった場合，国クラスに属する単語“ France ”と“ Angola ”をシードとして与えることで,「国と首都」や「国と言語」,「国と地域」など，国クラスと関わる単語クラスとその関係を得ることができる．また，この過程で Davidov らはある単語クラスに属する単語インスタンスを取得するために S_iHS_j に対する S_jHS_i のような対称的なパターンを用いた（S_i, S_j はある単語クラスに属する単語で，H は Davidov らが用いたコーパスで頻度が上位 100 件以内の単語を表す）．一方，本論文では単語クラスのインスタンスではなく，関係インスタンスの獲得を目的として対称的なパターンを用いることを提案している．また，Davidov らは，2 つの対称性しか利用していないのに対し，本論文では，複数の対称性 (表 2 参照) を考慮し，関係検索において最も有効な対称性の発見手法を提案している．さらに，関係類似度における対称性を定義している点でも Davidov らの研究とは異なる．これらのことか

ら，本手法は，Davidov らの既存研究と比べて新規性があると考えられる．

Kato ら [Kato 09] は，既存のウェブ検索エンジンを利用し，単語間の関係を bag-of-words モデルで表現し，関係検索を実現した．Kato らの手法では，まずペア (A, B) における A と B の関係を表す単語や語彙パターンの集合 T を抽出する．T は，ウェブ検索エンジンと χ^2 検定を利用して，A と B が含まれるページに偏って出現する単語や語彙パターンを抽出することで作成される．次に，C と，抽出された単語または語彙パターン $t \in T$ を使い，C と t のみとよく共起する単語を検索エンジンを使って抽出する．この単語が D の候補であり，χ^2 値の高いものほど上位にランキングされる．

Duc ら [Duc 10] は単語対のインデックスを作成して高速・高精度に関係検索を行う手法を提案した．語句間の関係は，本手法で用いたように，単語対の周辺の文脈からなるパターン文字列で表している．そして，単語対とパターン文字列からなる行列を作成し，それに対して Bollegala ら [Bollegala 09] によるクラスタリングアルゴリズムを適用することで同一の意味を持つ単語の異なる表現を一つの単語クラスタにまとめ，検索結果の精度および再現率を向上させた．なお，Kato らや Duc らの手法は本手法における $Q_{RS} = \{(A,B),(C,?)\}$，もしくは $Q_{RS} = \{(B,A),(?,C)\}$ というクエリのみを用いたものであり，8 種類の対称的なクエリおよび予測語の概念を定義し，それらを用いたスコア関数を提案したのは本論文が最初である．

7. むすび

本論文では，関係検索の性能を向上させるために，関係の対称性を用いた複数のクエリによるスコア関数を提案した．SAT データセットを用いた実験の結果，対称的なクエリを使わない場合に比べて F 値が 13.9% 向上した．また，SAT データセットを用いた関係類似度の先行研究と比較した場合，類義語シソーラスや次元圧縮および機械学習を用いない手法の中では最もスコアが高かった．これらのことから，本手法におけるスコア関数および対称性によるスコア補正が関係検索の性能を向上させていると言え，その有用性も高いと考えられる．

◇ 参 考 文 献 ◇

[Bicici 06] Bicici, E. and Yuret, D.: Clustering Word Pairs to Answer Analogy Questions, in *Proc. of TAINN'06* (2006)

[Bollegala 07] Bollegala, D., Matsuo, Y., and Ishizuka, M.: Measuring Semantic Similarity between Words using Web Search Engines, in *Proc. of WWW '07*, pp. 757–766 (2007)

[Bollegala 08] Bollegala, D., Matsuo, Y., and Ishizuka, M.: WWW sits the SAT: Measuring Relational Similarity on the Web, in *Proc. of ECAI'08*, pp. 333–337 (2008)

[Bollegala 09] Bollegala, D., Matsuo, Y., and Ishizuka, M.: Measuring the Similarity between Implicit Semantic Relations from the Web, in *Proc. of WWW'09*, pp. 651–660 (2009)

[Davidov 07] Davidov, D., Rapport, A., and Koppel, M.: Fully Unsupervised Discovery of Concept-Specific Relationships by Web Mining, in *Proc. of ACL'07*, pp. 232–239 (2007)

[Duc 10] Duc, N. T., Bollegala, D., and Ishizuka, M.: Using Relational Similarity between Word Pairs for Latent Relational Search on the Web, in *Proc. of the IEEE/WIC/ACM Int'l Conf. on Web Intelligence, WI'10*, pp. 196–199 (2010)

[Kato 09] Kato, M. P., Ohshima, H., Oyama, S., and Tanaka, K.: Query by analogical example: Relational Search using Web Search Engine Indices, in *Proc. of CIKM'09*, pp. 27–36 (2009)

[Mangalath 04] Mangalath, P., Quesada, J., and Kintsch, W.: Analogy-making as Prediction using Relational Information and LSA Vectors, in *Proc. of Int'l Conf. on Research in Computational Linguistics* (2004)

[Turney 03] Turney, P., Littman, M., Bigham, J., and Shnayder, V.: Combining Independent Modules to Solve Multiple-Choice Synonym and Analogy Problems, in *Proc. of RANLP'03*, pp. 482–486 (2003)

[Turney 05] Turney, P. and Littman, M.: Corpus-based Learning of Analogies and Semantic Relations, *Machine Learning*, Vol. 60, pp. 251–278 (2005)

[Turney 06a] Turney, P.: Expressing Implicit Semantic Relations without Supervision, in *Proc. of Coling/ACL'06*, pp. 313–320 (2006)

[Turney 06b] Turney, P.: Similarity of Semantic Relations, *Computational Linguistics*, Vol. 32, No. 3, pp. 379–416 (2006)

[Veale 04] Veale, T.: WordNet sits the SAT: A Knowledge-based Approach to Lexical Analogy, in *Proc. of ECAI'04*, pp. 606–612 (2004)

〔担当委員：高村 大也〕

2011 年 8 月 19 日 受理

著 者 紹 介

後藤 友和

2009 年名古屋工業大学工学部情報工学科卒，2011 年 東京大学大学院情報理工学系研究科創造情報学専攻修士課程修了．ウェブを用いた知識取得に興味を持つ．

グエン トアンドゥク（学生会員）

2007 年東京大学工学部電子情報工学科卒, 2009 年同大学院情報理工学系研究科創造情報学専攻修士課程修了，現在: 同専攻博士課程在学．ウェブからの情報抽出，ウェブ情報検索，並列分散プログラミングに興味を持つ．

ボレガラ ダヌシカ

2005 年東京大学工学部電子情報工学科卒．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 (短縮修了)．博士 (情報理工学)．現在: 同研究科・助教．複数文書自動要約，Web 上で人物の曖昧性解消，単語間の属性類似性，単語対間の関係類似性，Web からの関係抽出などの研究に興味を持つ．WWW, ACL, ECAI などの会議を中心に研究成果を発表．

石塚 満（正会員）

1971 年東京大学工学部卒，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 Perdue 大学客員准教授)．1992 年同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，電子情報通信学会，情報処理学会等の会員，本会の元会長．