1D1-4

高速・高精度ウェブ潜在関係検索エンジンの索引作成と関係表現手法

Relation representation and indexing method for fast and high precision latent relational Web search engine

グェン　トアン　ドゥク[*1]　ボッレーガラ ダヌシカ[*1]　石塚 満[*1]
Nguyen Tuan Duc　Danushka Bollegala　Mitsuru Ishizuka

[*1]東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

Latent relational search is a novel search paradigm based on proportional analogy between entity pairs. A latent relational search engine is able to return the word "Paris" as an answer to the question mark in the query {(Japan, Tokyo), (France, ?)}. We propose a method for extracting entity pairs from a text corpus to build the index for a high speed latent relational search engine. By representing the relation between two entities in an entity pair using lexical patterns, the proposed latent relational search engine can precisely measure the relational similarity between two entity pairs and can therefore accurately rank the result list. We have evaluated the system using a real world Web corpus and compare the performance with an existing relational search engine. The results show that the proposed method achieves high precision and MRR while requiring a small query processing time.

1. はじめに

潜在関係検索とは、単語ペア間の潜在的な関係を利用することにより入力単語ペアと類似する単語ペアを検索する新しい検索パラダイムである。潜在関係検索エンジンの概要は図 1 に示している。クエリー {(Tokyo, Japan), (?, France)} が入力されたときに、「Paris」を最初にランキングされた結果リストを返す。その理由はエンティティ・ペア「(Tokyo, Japan)」と「(Paris, France)」の関係類似度が高いからである。即ち、「Tokyo」と「Japan」との関係は「Paris」と「France」との関係が類似している (東京が日本の首都、パリもフランスの首都)。潜在関係検索のアイディアはアナロジー・シソーラスの

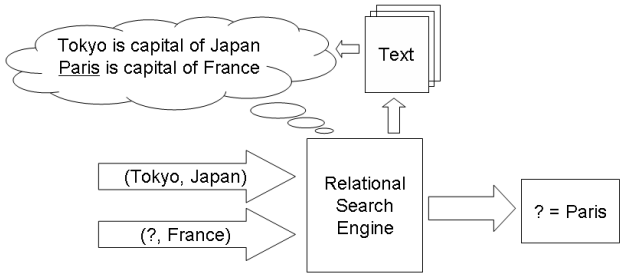

図 1: 潜在関係検索の例

研究 [Veale 03] や関係類似度測定研究 [Bollegala 09] で検討されてきた。潜在関係検索は自然言語処理、ウェブマイニングやレコメンダシステムなどの分野に対して応用可能性が高いである [Kato 09]。例えば、Turney が単語ペア間の関係類似度を利用し、類義語、上位語、下位語、対義語を自動的に見つけるための統一的な手法を提案した [Turney 08]。具体的な例で説明すると、"animal" の下位語を見つけるために、クエリー {(fruits, orange), (animal, ?)} を関係検索エンジンに問い合わせれば良いである。また、実際のアプリケーションで Apple ユーザが Microsoft のミュージック・プレイヤーを検索したい時に、クエリー {(Apple, iPod), (Microsoft, ?)} を検索すれば答え「Zune」が返ってくる [Kato 09]。つまり、キーワードを知らずに、情報を検索したい時に関係検索エンジンが効率的に使える。

本研究は [Bollegala 09] の研究成果を応用し、高速かつ精度の高い潜在関係検索を実現する方法を提案する。また、作成した検索エンジンを評価することにより、上記の検索パラダイムの実現可能性や実際に応用する可能性を明らかにする。

本稿は潜在関係検索の実現手法と評価結果をまとめる。以降、第 2. 節では、関連研究を紹介する。第 3. 節では潜在関係検索ためのエンティティ・ペアの抽出手法とエンティティ間の関係表現手法について説明する。また、第 4. 節で検索結果のランキング手法について述べる。提案手法の評価結果を第 5. で示す。最後に、第 6. 節でまとめと今後の課題について説明する。

2. 関連研究

関係類似度測定の研究 [Turney 06, Bollegala 09] では、単語間の関係を周辺文脈の語彙パターンで表し、パターン集合の類似度で関係類似度を定義する。本研究も同様に、関係を語彙パターンで表す。また、[Bollegala 09] で述べたパターンクラスタリング手法を使い、似ているパターンを一つのクラスタにまとめ、正確マッチングパターンの低頻度問題を解決する。

WWW2REL [Halskov 08] システムは、関係 R について、R(arg_1, ?) または R(?, arg_2) のようなクエリーに対して答えを出力することが出来る。WWW2REL はまず、関係 R を持つ 40 個の単語ペアをシード・ペアとして使い、関係 R を表現する語彙パターンを生成する。例えば、INDUCES という関係に対して、(carbon dioxide, headache) というペアから、特徴づける語彙パターンは "may cause", "lead to" などとして抽出する。次に、これらの語彙パターンを使い、クエリー INDUCES(aspirin, ?) に対して、"aspirin may cause *" (「*」はワイルドカードのオペレータで、多くの Web 検索エンジンでは 1 以上の単語にマッチする) などのクエリーをキーワード・ベース Web 検索エンジンに投げ、答え「apoptosis」を出力する。上記のように、WWW2REL は関係検索を実現

連絡先: グェン　トアン　ドゥク，東京大学大学院情報理工学系研究科創造情報学専攻，duc@mi.ci.i.u-tokyo.ac.jp

するが、潜在関係検索ではない。また、各関係について、40 個のシード・ペアを取得するために、シソーラスが必要であり、シソーラスに現れない関係に対しては答えを出せない。

Kato ら [Kato 09] は、既存のキーワード・ベース検索エンジンを利用し、単語間の関係を bag-of-words モデルで表現して、潜在関係検索を実現した。Kato らの手法は、関係検索のためにインデックスを作成せずに既存のキーワード・ベース Web 検索エンジンのインデックスを利用できるので、実装のコストが小さい。また、bag-of-words モデルを用いることで、幅広い範囲の単語種類を検索できるという利点がある。しかし、上記の手法は単語間における関係を十分に抽出できず、精度や平均逆順位 (MRR) がまだ低い。また、クエリー処理時にキーワード・ベース検索エンジンに数十個のクエリーを投げているので速度が遅い。

本研究では関係抽出の手法を使い、自動的に単語ペアや単語ペアの関係を特徴づける語彙パターンのインデックスを作成し、高精度の関係類似度測定 [Bollegala 09] の研究成果を応用し、高速かつ高精度の潜在関係検索を実現する。

3. エンティティ・ペア抽出と関係表現手法

3.1 エンティティ・ペアの抽出

本手法では、まず、テキスト・ドキュメント (Web ページ) を文に切り、文ごとを形態素解析器や固有表現抽出器に入れ、文を形態素に切り、かつ、固有表現を抽出する。文の解析結果の列から、自動的にエンティティ・ペアを抽出し、インデックスを作る。現在の実装では、固有表現だけを検索対象であるが、一般的に、どの単語種類でもペアのインデックスを作成したら検索できる。例えば、“It is now official: Microsoft acquires San Francisco based company Powerset for $100M.” という文から、三つのエンティティ・ペア (Microsoft, San Francisco), (Microsoft, Powerset), (San Francisco, Powerset) が抽出される。実際に関係を持っていないが、偶然に抽出されたペアをフィルタリングするために、出現頻度 5 以上のペアだけを検索対象にする。また、文中の距離が遠ければあまり関係を持たない可能性が高いので、距離がある閾値 M 以上のペアは検索対象としなく関係を抽出しない。

3.2 エンティティ間関係の表現

本研究は既存の関係類似度測定の研究 [Turney 06, Bollegala 09] に従い、エンティティ間の関係をエンティティが出現した周辺文脈の語彙パターンを使って表現する。語彙パターンはペアの出現位置の周辺語彙列を取り、列の n-gram として抽出する。ある文 S 中のエンティティ C と D の関係を表す語彙パターンを抽出するために、次の S 中の単語列 T を利用する:

$$b_1 b_2 ... b_k C w_1 w_2 ... w_m D a_1 a_2 ... a_p$$

即ち、単語列 T は C の前の k 語、語 C、C と D の間の単語列、語 D と D の後ろの p 語からなる単語列である (ここで、$m \leq M$ を満たす必要がある)。列 T から、すべての $n \leq (M+2)$ について、n-grams を生成する。生成された n-grams の中で、a_i だけまたは b_i だけを含む n-grams を捨てる。残りの n-grams 中、$w_i w_{i+1} ... w_j$ のような n-grams (w_i だけを含む n-grams) に対して、「$C * w_i w_{i+1} ... w_j * D$」に変形する (「*」 はワイルドカード記号で、ここでは 0 個以上の単語を表す)。また、D を含んでいない n-grams については、最後に「* D」を付ける。例えば、$b_k C w_1 w_2$ を $b_k C w_1 w_2 * D$ に置き換える。同様、C を含んでいない n-grams については、前に「C *」を付ける。得られた n-grams 集合の各 n-grams について、エンティティ C を変数 X に置き換え、エンティティ D を変数 Y に置き換える。更に、n-grams 中の各単語の語幹 (stem) を取り、最終の語彙パターン集合を作成する。例えば、文 “It is now official: Microsoft acquires San Francisco based company Powerset for $100M.” から、k = 3, p = 3 を設定する場合、次のような n-grams が生成される:

*X acquir * Y, X * San Francisco * Y, offici: X acquir * Y, now offici: X acquir San Francisco * Y, X * compani Y for $100M, X acquir San Francisco base compani Y, ...*

上記のように、本手法も従来研究 [Turney 06, Bollegala 09] と同じような語彙パターン抽出アルゴリズムを用いるが、検索の再現率を上げるために、二つの工夫点を加えた。一つ目は、C と D が同時に含まない語彙パターンも取る。これにより、各エンティティ・ペアのエンティティ間の単語列が正確にマッチングしなくても、類似する単語列があればそのエンティティ・ペアが類似すると認識できる。例えば、次の二つ文 “Obama is the 44th president of the U.S” と “Sarkozy is the current president of France” において、パターン「current president of」を取る (つまり、n-grams 「X * current president of * Y」を生成する) と、二つのペア (Obama, U.S) , (Sarkozy, France) が共有のパターンを持ち、類似度が高くなる。二つ目は、n-grams の各単語の語幹だけを取る。これにより、過去形、複数形などの違いを吸収でき、再現率を高めることができる。単語ペア (エンティティ・ペア) wp について、それと一緒に出現した語彙パターンの集合を $\mathbf{P}(wp)$ とする:

$$\mathbf{P}(wp) = \{p_1, p_2, \ldots, p_n\} \quad (1)$$

また、語彙パターンが一つ以上共有している単語ペアを高速に検索するために、ある語彙パターンがどの単語ペアと一緒に出現したかの情報を転置インデックスに保存する。$\mathbf{W}(p)$ を語彙パターン p と一緒に出現した単語ペアとする:

$$\mathbf{W}(p) = \{wp_1, wp_2, \ldots, wp_n\} \quad (2)$$

また、単語ペア wp_i が語彙パターン p_j で出現した頻度を $\mathrm{f}(wp_i, p_j)$ とする。その時、語彙パターン p の単語ペア頻度ベクトル $\mathbf{\Phi}(p)$ を次のように定義する:

$$\mathbf{\Phi}(p) = (\mathrm{f}(wp_1, p), \mathrm{f}(wp_2, p), \ldots, \mathrm{f}(wp_n, p))^{\mathrm{T}} \quad (3)$$

同様、単語ペア wp の語彙頻度ベクトルを次のように定義する:

$$\mathbf{\Psi}(wp) = (\mathrm{f}(wp, p_1), \mathrm{f}(wp, p_2), \ldots, \mathrm{f}(wp, p_n))^{\mathrm{T}} \quad (4)$$

3.3 語彙パターン・クラスタリング

語彙パターンの各単語の語幹を取ったとしても、二つの単語ペアで全く一致する語彙パターンを共有する確率がまだ低く、検索の再現率が低い。そこで、意味が類似する語彙パターンをクラスタリングすることにより、完全マッチングでなくても、意味が似ていれば、同じ語彙パターンと見なし、類似度を測定することで、再現率を上げることができる。

語彙パターンをクラスタリングできるために、二つの語彙パターンの類似度を定義する必要がある。本研究も従来研究 [Bollegala 09] と同様、語彙パターンの単語頻度ベクトルのコサイン類似度を使い、語彙パターンの類似度を定義する。

語彙パターンのクラスタリングは、[Bollegala 09] で述べた逐次クラスタリング・アルゴリズムを利用する。考えている語彙パターンについて、そのパターンとの類似度がある閾値 θ 以上のパターン・クラスタが存在すれば、そのパターンは当該するクラスタに追加される。それ以外の場合、そのパターン自身が一つの新しいクラスタを形成する。アルゴリズムの詳細は [Bollegala 09] に参考されたい。

3.4 エンティティ・クラスタリング

一つのエンティティが複数の表現形を持つことが多い。例えば、“United States” はよく “U.S”, “U.S.” などの形で略記されるし、“America” で置き換えることもある。これらの複数表現形を吸収し、同じエンティティの複数表現形を統一に扱えるために、本研究はエンティティをクラスタリングする手法を提案する。あるエンティティの特徴ベクトルとして、そのエンティティとペアをなす相手のエンティティの頻度を利用する。また、エンティティの類似度を相手エンティティ頻度ベクトルのコサイン類似度として定義する。エンティティのクラスタリングも上記と同様、エンティティ・クラスタリング類似度閾値 ξ を定義し、[Bollegala 09] で述べた逐次クラスタリング・アルゴリズムを使用する。

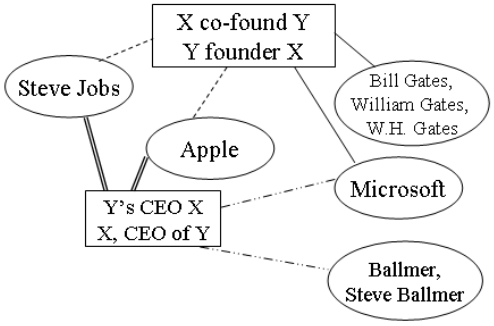

図 2: エンティティと語彙パターンの関係

パターン・クラスタリングとエンティティ・クラスタリングを行った後、図 2 で示すように、同一のエンティティの複数表現形が一つのエンティティ・クラスタにまとめられ、エンティティ・クラスタ間関係が語彙パターン・クラスタを経由して表されている。

4. 候補の検索とランキング

4.1 候補の検索

クエリー {(A, B), (C, ?)} に対して、その答えの候補は (A, B) との頻度 10 以上の語彙パターンを一つ以上共有している頻度 5 以上の (C, X) 形を持つエンティティ・ペア集合 $\Re$ からで検索する:

$$\Re = \bigcup_{p \in \mathbf{P}(s) \wedge \mathrm{freq}(p) \geq 10} \{wp \in \mathbf{W}(p) | (wp[0] = C) \wedge \mathrm{freq}(wp) \geq 5\} \tag{5}$$

4.2 候補のランキング

候補をランキングするために、エンティティ・ペアの関係類似度を計算する必要がある。関係類似度を計算するときに、語彙パターンのクラスタ情報を考慮し、同じクラスタにある語彙パターンが同じパターンとして扱う。候補ペア c とクエリーで入力されたペア s の関係類似度は Algorithm 1 で計算する。

ランキングための候補集合 Γ は s との関係類似度がある閾値 σ 以上のペアの集合である:

$$\Gamma = \{c \in \Re | \mathrm{relsim}(s, c) \geq \sigma\} \tag{6}$$

また、クエリー {(A, B), (C, ?)} に対して、上記の候補検索プロセスをその逆クエリー {(B, A), (?, C)} で行い、最終の候補のスコアを計算する。 $s' = (B, A)$ 、 $c' = (X, C)$ とすると、候補 c は質問ペア s に対して、最終スコアは

$$\chi(s, c) = \mathrm{relsim}(s, c) + \frac{1}{2}\mathrm{relsim}(s', c') \tag{7}$$

として定義する (ここで、オリジナルのクエリーから得られたスコアを優先するので、逆クエリーで得られたスコアの重みを 1/2 とする)。最後に、エンティティ・クラスタの情報を使い、結果クラスタのスコアを計算する。候補クラスタ K ($K = \{c_1, c_2, \ldots, c_k\}$) のスコアは次のように定義する:

$$\mathrm{score}(s, K) = \frac{1}{k}\sum_{i=1}^{k} \chi(s, c_i) \tag{8}$$

最終の結果リストは候補クラスタのスコアをでランキングされたものである。

Algorithm 1 relsim(s, c)

Input: two word pairs s and c
Output: relational similarity between s and c

```
// Initialize inner product to 0
ρ ← 0
// Initialize set of used patterns
℘ ← {}
for pattern p ∈ P(c) do
  if p ∈ P(s) then
    ρ ← ρ + f(s,p)f(c,p)
    ℘ ← ℘ ∪ {p}
  else
    Ω ← the cluster that contains p
    max ← −1
    q ← null
    for pattern p_j ∈ (P(s)\P(c))\℘ do
      if (p_j ∈ Ω) ∧ (f(s,p_j) > max) then
        max ← f(s,p_j)
        q ← p_j
      end if
    end for
    if max > 0 then
      ρ ← ρ + f(s,q)f(c,p)
      ℘ ← ℘ ∪ {q}
    end if
  end if
end for
return ρ/(‖Ψ(s)‖‖Ψ(c)‖)
```

5. 評価

パラメータ調整をするために、12000 ウェブページのテキストコーパスを使った。また、システムの性能を評価するために、前述とは別に、6000 個のウェブページのテキストを使った。これらのテキストには主に 4 つの種類の関係が含まれている: 人の生まれ場所 (Einstein - Germany), 会社の本社所在地 (Microsoft, Redmond), 会社の社長 (Eric Schmidt - Google) と会社買収関係 (Google - Youtube)。上記のテキスト・コーパスから 113742 個の単語ペアが抽出された (その内、4103 ペアが頻度 5 以上)。また、抽出された語彙パターン数は 2069121 である。テストのクエリー・セットは 842 クエリーがあり、その内、12 個のクエリーが複数正解 (ある会社が買収した会社集合) を持つ。正解が 1 つしかない場合、トップ 1 結果だけの精度と再現率を評価し、正解が複数の場合、トップ 10 の結果を評価する。

5.1 エンティティ・クラスタリング閾値の調整

検索の精度を高めるために、エンティティ・クラスタリング・アルゴリズムの精度をできるだけ高くする必要がある。エンティティ・クラスタリングの類似度閾値 ξ を変化しながら、精度を測ったところ、ξ が 0.3 以上の時、クラスタリング精度が 100%であった。ξ が大きくなると、正解クラスタの再現率が減るので、精度 100%で、最大の再現率を出す閾値は $\xi = 0.3$ である 。そこで、以降の実験では、ξ を 0.3 に設定して行う。

5.2 語彙パターン・クラスタリングの類似度閾値の影響

語彙パターン・クラスタリング類似度閾値 θ の最適値を探すために、θ を変化しながら実験を行い、検索 F-score を測った。正解数が複数の場合、再現率が測定できないので、F-score

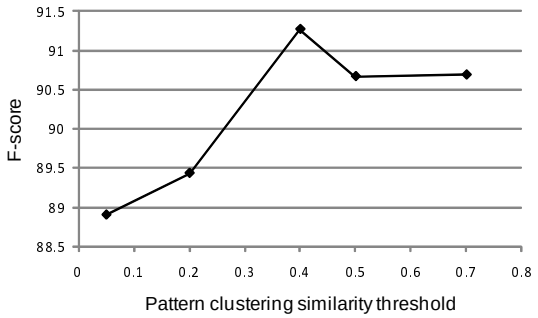

図 3: 三つの関係種類 (人の生まれ場所, 会社本社在地, 会社の社長) の平均 F-score とパターン・クラスタリング閾値 θ の関係 ($\xi = 0.3$, $\sigma = 0.05$ の時)

を計算できない。そこで、再現率が計算できる 3 つの関係種類 (人の生まれ場所、会社本社在地、会社の社長) で平均 F-score を測った。図 3 は上記の 3 つの関係種類の平均 F-score とパターン・クラスタリング閾値 θ の関係を表している。この図から分かるように、θ が 0.4 の時、最大の F-score が得られた。また、候補検索ための類似度閾値 σ を [0.03, 0.2] の間で変化しながら、各 θ の値で平均精度と平均 F-score を測定したところ、上記の図の形が変わらず、θ が 0.4 の時、ピークが得られた。ただし、σ が 0.05 の時に、最大の F-score が得られた。σ が 0.2 よりも大きくなると、検索の再現率が極めて小さいので、F-score が小さくなる。従って、θ が 0.4 と σ が 0.05 の時、検索エンジンの性能がもっともよいである。

5.3 平均精度、再現率と F-score

上記の最適のパラメータ ($\theta = 0.4, \xi = 0.3, \sigma = 0.05$) で本システムの平均精度、再現率と F-score を測定した (再現率と F-score は正解が 1 つの関係種類だけを測定した)。使用データセットは 6000 個のウェブ・ページで、パラメータ決定の時のデータセットと違うデータセットである。これは、パラメータ調整で最適なパラメータの値を導いたが、そのデータセットに固有的なパラメータの値である可能性があるから、別のデータセットで測定し、このバイアスを防ぐのである。実験結果を表 1 で示してる。

表 1: 本検索エンジンの性能 ($\theta = 0.4, \xi = 0.3, \sigma = 0.05$ の時)

Data set	Precision	Recall	F-score
Birthplace	98.89	98.89	98.89
Headquarters	90.59	85.56	88.00
CEO-comp.	95.56	95.56	95.56
Acquirer - Acquiree	81.34	-	-
Average	91.60	93.34	94.15

5.4 既存関係検索エンジンとの比較

本節では、Kato らが提案した関係検索システム [Kato 09] との比較結果を示す。比較データとしては、上記の 6000 個の (英語) ウェブ・ページのデータセットにおける本システムの性能と、Kato ら [Kato 09] で示した性能である。この比較は検索対象の言語 (英語と日本語) が異なると関係種類が異なるため、あまり公平と正確でないが、共通の関係種類もいくつかあるので、ここで性能比較を行う。表 2 は比較結果を示してい

表 2: Kato らの関係検索との比較 (@N は Top N 結果に正解があるクエリーの比率).

Method	MRR	@1	@5	@10	@20
[Kato 09]	0.545	43.3	68.3	72.3	76.0
Proposed method	0.963	95.0	97.8	97.8	97.8

る。この表では、MRR は平均逆順位で、高ければ高いほど性能がよい (最大値が 1 である)。また、@N はトップ N 結果中に正解がある比率を表している。表 2 から分かるように、本システムは Kato らのシステムよりもよい性能を出している。また、本検索エンジンのクエリー処理時間は 10 秒以内であり、実践応用の処理時間であると考えられる。

6. おわりに

本稿では、エンティティ・ペア抽出とインデックシング手法を提案し、正確な関係類似度測定の研究成果を応用して、高速・高精度の潜在関係検索エンジンを実現した。今後はもっと膨大なウェブ・コーパスを使い、検索を実現し、実際の応用ができるようにする予定である。

参考文献

[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, ACM (2009)

[Halskov 08] Halskov, J. and Barriere, C.: Web-based extraction of semantic relation instances for terminology work, *Terminology*, Vol. 14, No. 1, pp. 20–44 (2008)

[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)

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

[Turney 08] Turney, P. D.: A uniform approach to analogies, synonyms, antonyms, and associations, in *Proc. of Coling'08*, pp. 905–912 (2008)

[Veale 03] Veale, T.: The Analogical Thesaurus, in *Proc. of the Innovative Applications of Artificial Intelligence*, pp. 137–142, AAAI Press (2003)