2010 年大会速報論文特集

Short Paper

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 a Fast and High Precision Latent Relational Web Search Engine

グェン トアン ドゥク
Nguyen Tuan Duc
東京大学情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
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

石塚 満
Mitsuru Ishizuka
（同　上）
ishizuka@i.u-tokyo.ac.jp

keywords: latent relational search, relational similarity, analogical search, relational web search

Summary

Latent relational search is a new search paradigm based on the proportional analogy between two entity pairs. A latent relational search engine is expected to return the entity "Paris" as an answer to the question mark (?) in the query {(Japan, Tokyo), (France, ?)} because the relation between Japan and Tokyo is highly similar to that between France and Paris. We propose a method for extracting entity pairs from a text corpus to build an 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 evaluate the system using a 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 small query processing time. In particular, the proposed method achieves an MRR of 0.963 and it retrieves correct answer in the Top 1 result for 95% of queries.

1. ま え が き

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 {(Tokyo, Japan), (?, France)} が入力されたときに，"Paris" が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)．

自然言語処理，Web マイニングやレコメンダシステムなどの分野において関係検索が応用できる可能性があり，関係検索システム実現への期待が存在する [Kato 09]．例えば，"animal" の下位語を見つけるために，クエリ {(fruits, orange), (animal, ?)} を関係検索エンジンに問い合わせれば良い．また，製品名検索の例を挙げると，Apple の iPod ユーザが Microsoft の相当するミュージックプレイヤーを検索したい時，クエリ {(Apple, iPod), (Microsoft, ?)} で検索を行えば，"Zune" が答えとして得られる [Kato 09]．このように，キーワードを知らずに，情報を検索したい時に潜在関係検索エンジンが効率的に使える．つまり，未知の領域での検索を行いたい時，潜在関係検索が有効であるといえる．これは，潜在関係検索エンジンが異なる領域間の知識マッピング能力を持つからである．即ち，ユーザの既知の領域 (Apple の製品) における知識を，未知の領域 (Microsoft の製品) にマッピングすることで，未知の領域の知識を獲得する，という能力である． 本研究では我々が考案した関係類似度計算の手法 [Bollegala 09] 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, 提案したシステムをウェブコーパスで評価し, 既存の関係検索システムと比較し, 提案手法が高精度, 高速かつ高い平均逆順位 (mean reciprocal rank, MRR) を達成できることを示す. これらの実績により, 潜在関係検索が実践的な応用レベルでの実現可能性を示す.

以降，第 2 章では，関連研究を紹介する．第 3 章では

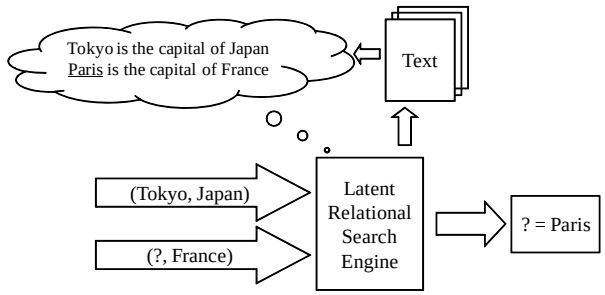

図 1 潜在関係検索の例

潜在関係検索のためのエンティティペア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 章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 章で示す．最後に，第 6 章でまとめと今後の課題について説明する．

2. 関 連 研 究

関係類似度計算の研究 [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，完全パターンマッチングの低頻度問題を解決する．しかし，上記の研究は単語ペアの類似度計算に関する研究であるため，4 つの単語が既に与えられたと仮定しているので，潜在関係検索を実現していない．

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 Index 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 Index 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を投げているので速度が遅い．

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，与えられたテキストコーパス中の各テキストドキュメント (Web ページなど) を文ごとに区切り，各文を形態素解析器や固有表現抽出器 *1に入れ，文を形態素に分け，また固有表現を抽出する．文の解析結果の中で 2 つ以上のエンティティがあれば，文中の前後順序を保ったすべてのエンティティのペアを抽出する．ここで注意すべき点は，エンティティペアの関係種類が事前に分かる必要はなく，すべてのペアを抽出するということである．例えば，“It is now official: Microsoft acquires San Francisco based company Powerset for $100M.” という文からは，3 つのエンティティペア (Microsoft, San Francisco), (Microsoft, Powerset), (San Francisco, Powerset) が抽出される．ここで，実際には有意な関係を持っていないにもかかわらず，偶然抽出されたペアをフィルタリングするために，コーパス解析結果で出現頻度 5 以上のペアだけを検索対象として扱う．また，文中での距離が遠いエンティティ同士は明確な関係を持たない可能性が高いので，その間の距離がある閾値 M よりも大きいエンティティペアは検索対象とせず，抽出しない．

*1 Stanford POS Tagger と Stanford Named Entity Recognizer を使用している． http://nlp.stanford.edu/software/CRF-NER.shtml

3·2 エンティティ間の関係の表現

本研究では先行研究 [Turney 06, Bollegala 09] に従い，エンティティ間の関係をそれらのエンティティが出現した文の周辺文脈を考慮して表現する．具体的には，エンティティペアにおける意味関係を，それぞれエンティティで挟まれる語彙パターン (lexical pattern) と，それらの前後に出現する語彙パターンの頻度で表現する．語彙パターンの抽出対象を各エンティティの前後を含むエンティティペアの周辺の単語列 (文のサブシーケンス) だけにするのは，遠く離れた語彙パターンはエンティティと関係の薄いものである可能性が高いからである．また，文全体を抽出対象にすると，語彙パターンの数は膨大になる傾向がある．形態素解析と固有表現抽出を行った文 S におけるエンティティC と D の関係を特徴づける語彙パターンを抽出するために，まずエンティティ C を変数 X，エンティティD を変数 Y に置き換える．これは，語彙パターンを特定のエンティティペアに依存しないように (あらゆるエンティティペアで共有できるように) 表現する必要があるからである．更に，単語の語形変化 (活用など) の違いを吸収するために，文中の各単語を stemming する．次に，以下の S の部分単語列 T を調べる:

$$T = b_1 b_2 ... b_k X w_1 w_2 ... w_m Y a_1 a_2 ... a_p$$

単語列 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，“$X * w_i w_{i+1} ... w_j * Y$” 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，Y を含んでいない n-grams については，最後に “$*Y$” を付ける．例えば，$b_k X w_1 w_2$ を $b_k X w_1 w_2 * Y$ に置き換える．同様，X を含んでいない n-grams については，前に “$X*$” を付ける．

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，次の工夫点を加えている．まず第一の工夫として，語彙パターンの中の単語を stemming し活用形の違いを吸収する．これにより類似度計算の際，違う活用形を持った語彙パターンが同一に扱われ，再現率を高くすることができる．また第二の工夫として，n-gram は X と Y 両方を含むという条件を省く．その代わり，“X∗” や “∗Y” を付加することで，語彙パターンとエンティティとの相対位置を区別する．これにより，2 つのエンティティペアが共通の語彙パターンを持つ確率が高くなるので，検索の再現率を向上できる．例えば，“Obama is the 44th and current president of the U.S.” と “Sarkozy is the current president of France” という文において，もし語彙パターン “current president of” を許可 (つまり語彙パターン “X* current president of * Y” を生成) すると，(Obama, U.S) と (Sarkozy, France) が共通の語彙パターンを持つようになる．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 Index 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:

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

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:

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

同様，単語ペア wp の語彙パターン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とエンティティクラスタリング

語彙パターンの各単語の語幹を取ったとしても，2 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，類似度を計算することで，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ようにするため，2 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 p と q の類似度を p と q の単語ペア頻度ベクトルのコサイン類似度 $\cos(\mathbf{\Phi}(p), \mathbf{\Phi}(q))$ として定義する．

語彙パターンのクラスタリング (以降，パターンクラスタリングとも呼ぶ) 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．パターンクラスタリングの結果として，類似意味を持つ語彙パターンの語彙パターンクラスタ (パターンクラスタとも呼ぶ) が生成される．

また，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”, “U.S.”, “America”)．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的に扱えるようにするため，本研究はエンティティをクラスタリングする手法 (以降，エンティティクラスタリングと呼ぶ) を提案する．2 つのエンティティが同じようなエンティティ集合と一緒にペアをなすなら，それらのエンティティが似ているコンテキストで出現することが分かる．この場合，分布仮説 (distributional hypothesis) により，その 2 つのエンティティが類似する．それ故，2 つのエンティティ間の類似度は，それぞれのペアとなるエンティティ (以降，パートナーエンティティと呼ぶ) の頻度ベクトル同士のコサイン類似度で定義される．即ち，エンティティペア (w_i, w_j) の出現頻度を $\mathrm{f}(w_i, w_j)$ とし，エンティティ w_j のパートナーエンティティの頻度ベクトルの i 番目を $h(w_j, i) = \mathrm{f}(w_j, w_i) + \mathrm{f}(w_i, w_j)$ とする．次に，エンティティ C と D それぞれのパートナーエンティティの頻度ベクトル同士のコサイン類似度は，以下の式で計算する．

$$\mathrm{simW}(C, D) = \frac{\sum_i h(C, i) \cdot h(D, i)}{\sqrt{\sum_i h^2(C, i)} \sqrt{\sum_i h^2(D, i)}} \quad (5)$$

エンティティのクラスタリングも上記と同様に，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．エンティティクラスタリングの結果として，類似するエンティティのエンティティクラスタが生成される．

4. 候補の検索とランキング

4·1 候 補 の 検 索

クエリ {(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 $\Re$ として定義する．ここで，ソースペアを s ($s = (A, B)$) とし，$\mathrm{freq}(wp)$ をエンティティペア wp の頻度とすると，

$\Re$ は以下のように計算される．

$$\Re = \bigcup_{p \in \mathbf{P}(s) \wedge \mathrm{freq}(p) \geq 10} \{wp \in \mathbf{W}(p) | (wp[0] = C) \wedge \mathrm{freq}(wp) \geq 5\} \quad (6)$$

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を同じパターンとして扱う．候補ペア c ($c = (C, X)$) とクエリで入力されたペア s ($s = (A, B)$) の関係類似度は Algorithm 1 で計算する．Algorithm 1 では，ペア s と c の間の関係類似度 $\mathrm{relsim}(s, c)$ は，語彙パターンの頻度ベクトルの擬似コサイン類似度として定義される．擬似内積 $\mathbf{\Psi}(s) \cdot \mathbf{\Psi}(c)$ は次のように計算する．まず，語彙パターン p が $\mathbf{P}(s) \cap \mathbf{P}(c)$ に属すならば，普通の内積と同様に，$\mathrm{f}(s, p) \cdot \mathrm{f}(c, p)$ を内積に加える．パターン p が $\mathbf{P}(c)$ に属しているが，$\mathbf{P}(s)$ には属していない場合，$\mathbf{P}(s) \backslash \mathbf{P}(c)$ (差集合) に属し，かつ p と同じ語彙パターンクラスタに属すパターン q を見つける．もし複数の q が存在していれば，出現頻度が最大のものを選ぶ．選んだパターン q は，p と同じクラスタに属するため p と意味的に類似するので，$\mathrm{f}(s, q) \cdot \mathrm{f}(c, p)$ を内積に加える．また，選んだ q を以降の内積計算プロセスから除外するため，マークしておく (Algorithm 1 では q を $\mathbf{T}$ に追加する)．

ランキングのための候補集合 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:

$$\Gamma = \{c \in \Re | \mathrm{relsim}(s, c) \geq \sigma\} \quad (7)$$

また，クエリ {(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 {(B, A), (?, C)} で行い，最終の候補のスコアを計算する．逆クエリのスコアも利用するのは，関係類似度がエンティティペアの転置操作に対して不変であるという性質があるからである [Goto 10]．ここで，$s' = (B, A)$ ，$c' = (X, C)$ とすると，候補 c は質問ペア s に対して，最終的なスコアは

$$\chi(s, c) = \mathrm{relsim}(s, c) + \frac{1}{2}\mathrm{relsim}(s', c') \quad (8)$$

として定義する (ここで，オリジナルのクエリから得られたスコアを優先するので，逆クエリ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．候補エンティティペア $c_i = (C, X_i)$ とすると，候補エンティティのクラスタ K ($K = \{X_1, X_2, \ldots, X_k\}$) のスコアは次のように定義する:

$$\mathrm{score}(s, K) = \frac{1}{k}\sum_{i=1}^{k} \chi(s, c_i) \quad (9)$$

最終的な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 entity pairs s and c
Output: the relational similarity between s and c

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

表 1 評価用の関係

関係	ペアの数	エンティティペアの例
人-出身地	20	(Franz Kafka, Prague), (Andre Agassi, Las Vegas), (Charlie Chaplin, London)
会社-本部地	15	(Google, Mountain View), (Microsoft, Redmond), (Apple, Cupertino)
社長-会社	16	(Eric Schmidt, Google), (Steve Ballmer, Microsoft), (Carol Bartz, Yahoo)
Acquirer - Acquiree	48	(Google, Youtube), (Microsoft, Powerset), (Yahoo, Kelkoo)

5. 実験による評価

5·1 データセット

提案手法のパラメータ調整や評価のために，表 1 に示した 4 種の関係を使い，パラメータの値を変化させながら検索エンジンの性能を測定した．これらの関係は，関係抽出や関係類似度計算に関する研究でよく用いられるものである [Bollegala 09, Banko 08, Bunescu 07]． システムの性能を評価するためには，上記の 4 種の関係のインスタンスを含んだテキストコーパスを用意する必要がある．一般的に，このテキストコーパスには評価対象になる関係 (上記の 4 種の関係) とは関連しないドキュメントも含まれている．例えば，コーパスには 表 1 で示した関係の情報の他， Web ページ上の広告のテキストやノイズ的テキストがよく入っている．それだけでなく， 評価対象の関係情報とは全く関連しない記事などを含むこ

とも多い．しかしこれらの場合にも，提案システムは正しくエンティティペアやペアの語彙パターンを抽出できる．一方で，評価対象と関連しない記事の割合が大きくなると，評価に必要な関連記事を十分な数にするためには，膨大な量のコーパスが必要となる．またその時，エンティティペアや語彙パターンが膨大な数となり，パラメータを変化する度に長時間を費やすため，評価にかかる時間とコストが大きくなる．そこで，ドキュメントを評価対象の関係情報を含んだものに限定するために， 表 1 にある関係の情報 (関係を表すキーワード) とそのインスタンス (エンティティペア) を使い，Google [*2]にクエリを投げ，各クエリの検索結果から上位 100 件の URL を集めた．これらの URL にある Web ページをダウンロードし，評価用のテキストコーパスを作成する．

パラメータ調整用のコーパスには，12000 個のドキュメント (Web ページ) が入っている．これらのドキュメントには， 表 1 におけるエンティティペアや関係の関連情報が入っているが，関連エンティティの数や関係の数は正確には特定できない．なぜなら，例えば Microsoft による Powerset の買収の記事の中に，Apple による Emagic の買収の情報も入っている，といったことが起こり得るからである ((Apple, Emagic) というペアが 表 1 に入っていない場合にも (Apple, Emagic) ペアが抽出される)．

集められたドキュメント集合から，HTML タグをすべて削除し，テキスト内容を取得し，エンティティペアと語彙パターンの抽出を行う．その解析結果として，113,742 個のエンティティペアと 2,069,121 個の語彙パターンが抽出された．

パラメータ調整用のテストクエリセットは 842 クエリあり，その内，12 個のクエリが複数正解 (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．

5·2 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 を変化させながら，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．

5·3 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 を変化させながら実験を行い，検索結果の F-score を測った． 正解数が複数の場合では，以前説明したように，コーパスの中で関連エンティティがどの程度あるかは特定できず，評価の際，検索結果のトップ

*2 http://www.google.com

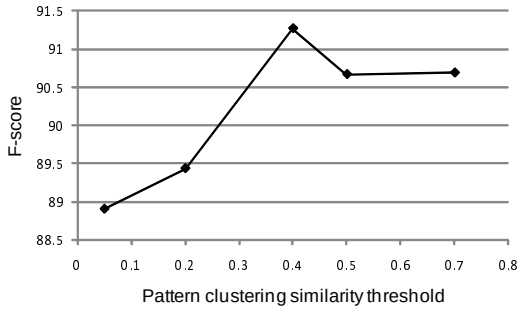

図 2 三つの関係種類 (人の生まれ場所, 会社本社在地, 会社の社長) の平均 F-score とパターンクラスタリング閾値 θ の関係 (ξ = 0.3, σ = 0.05 の時)

表 2 本検索エンジンの性能 ($\theta = 0.4, \xi = 0.3, \sigma = 0.05$ の時) (Acquirer-Acquiee の関係については，上記で説明したように，再現率が計算できない)

クエリセット	精度	再現率	F-score
人-出身地	98.89	98.89	98.89
会社-本部地	90.59	85.56	88.00
会社-社長	95.56	95.56	95.56
Acquirer-Acquiree	81.34	-	-
平均	91.60	93.34	94.15

10 しか考慮しないので，再現率が測定できなく，F-score も計算できない．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 2 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．σ については 0.05 の時に最大の F-score が得られ，0.2 よりも大きくなると検索の再現率が極めて小さいとなり，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くなることから，この値に設定した．

5·4 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， 表 1 で示した 4 種の関係を含むが，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り，関係のインスタンス (エンティティペア) も異なる．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，使用した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ことから，別のデータセットで測定し，このバイアスを防ぐためである．実験結果を 表 2 で示してる．表 2 から分かるように，提案手法は正解が 1 つ持つクエリセットの Top 1 結果は 90%以上が正解である．

表 3 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の比率).

手法	MRR	@1	@5	@10	@20
KatoTC-JA [Kato 09]	0.379	25.0	55.3	60.3	67.3
KatoTC-EN	0.332	20.0	50.0	65.6	71.1
KatoCNJ-JA [Kato 09]	0.545	43.3	68.3	72.3	76.0
提案手法 (Prop-EN)	0.963	95.0	97.8	97.8	97.8

5·5 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．Kato ら [Kato 09] で示した実験結果は日本語での結果である．一方，本研究は英語のテキストで評価している．そこで，我々は Kato らが述べた手法の一部を英文のテキストに適用できるように再実装し，5·4 節に説明したクエリセットで評価を行い，比較をする．再実装した手法は Kato ら [Kato 09] の Term Co-occurence (KatoTC) による手法である．KatoTC 手法ではエンティティペアの関係を bag-of-words モデルで表現する．即ち，ペア (A, B) の A と B との関係は，A と B 両方を含むドキュメントにしかよく出現しない単語の集合で表す．一方，提案手法は語彙パターンを使い，関係を表現する．そこで，2 つの手法の性能を同じクエリセットで比較することにより，語彙パターンによる手法と共起単語による手法の優劣を明らかにすることができる．また，Kato らでの最良結果を出しているのは KatoCNJ 手法である．KatoCNJ 手法のランキングは KatoTC と KatoSP(syntactic pattern を利用する手法) におけるランキングの 2 次式の組み合わせである．従って，再実装した KatoTC 手法の性能から，Kato らで述べた KatoCNJ と KatoTC の性能の比を使い，我々のクエリセットにおける KatoCNJ の性能を推定することができると考えられる．これらの実験は正解を 1 つ持つクエリ種類について行う．

表 3 は比較結果を示している．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．KatoTC-JA, KatoCNJ-JA は Kato ら [Kato 09] で述べた手法の和文のテキストに対する結果である (KatoCNJ-JA は Kato らが示した最良結果である)．KatoTC-EN は我々が実装した KatoTC 手法を提案手法と同じクエリセットで評価した時の評価結果である．KatoCNJ-JA と KatoTC-JA の MRR 比は 1.44 (0.545/0.379) であり，もしこの比が我々のクエリセットにも適用できたら，KatoCNJ-EN の MRR の値が KatoTC-EN の値から，0.477 であると推定される．従って，提案手法 (Prop-EN) は KatoCNJ-JA や推定された KatoCNJ-EN 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処理時間は 10 秒以内であり，実用の処理時間であると考えられる．キーワードベース検索エンジンをクエリする潜在関係検索システム [Kato 09, Goto 10] では，この高速なクエリ処理時間を達するのが難しいと考えられる．

6. むすび

本稿では，エンティティペア抽出と Indexing 手法を提案し，正確な関係類似度計算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．今後はより膨大なウェブコーパスを使い検索を実現し，実用性を高める予定である．

◇ 参 考 文 献 ◇

[Banko 08] Banko, M. and Etzioni, O.: The Tradeoffs Between Open and Traditional Relation Extraction, in *Proc. of ACL'08*, pp. 28–36 (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, ACM (2009)

[Bunescu 07] Bunescu, R. C. and Mooney, R. J.: Learning to Extract Relations from the Web using Minimal Supervision, in *Proc. of ACL'07*, pp. 576–583 (2007)

[Goto 10] Goto, T., Duc, N., Bollegala, D., and Ishizuka, M.: Exploiting Symmetry in Relational Similarity for Ranking Relational Search Results, in *Proc. of PRICAI'10*, pp. 595–600 (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)

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

〔担当委員：土田 正明〕

2010 年 7 月 20 日 受理

著 者 紹 介

グェン 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，単語ペア間の関係類似性，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．