高速な類似度計算手法による関係パタンのクラスタリング

高瀬翔† 岡崎直観 †‡ 乾健太郎†
東北大学† 科学技術振興機構さきがけ‡
{takase, okazaki, inui}@ecei.tohoku.ac.jp

1 はじめに

単語間の意味的関係についての知識は，情報検索や推論，質問応答など自然言語処理応用のために必要不可欠である [5, 6]．例えば意味的関係として，is-a⟨ にんにく, 食物 ⟩，prevent⟨ にんにく，癌 ⟩ という知識を持っていたとすると,「癌に効く食物は何か？」という問いに対し,「にんにく」と答える事ができる．この「is-a」や「prevent」は意味的関係の種類を表しており,⟨ 食物，にんにく ⟩ や ⟨ にんにく，癌 ⟩ のように特定の意味的関係にある単語対を関係インスタンスと呼ぶ．柔軟な推論システムや，いかなる質問にも答えられる質問応答システムを構築するためには，ありとあらゆる関係インスタンスを獲得しておく必要がある．

関係インスタンスを獲得するためには，テキスト中で関係を表す表現（関係パタン）を認識する必要がある．近年，人手での事前知識なしにテキストに書かれているあらゆる関係パタン，関係インスタンスの獲得を自動で行うという，**Open Information Extraction（Open IE）**が盛んに研究されている [4]．Open IE システムの研究は，英語を対象に行っているものが多く，それ以外の言語を対象としたものはあまり見られない．多くの Open IE システムでは，名詞間の単語列を関係パタンとしている [1, 4]．これは英語のような語順の制約が強い言語では有用であるが，日本語のような語順の自由度が高い言語には適当でない．例えば，名詞間の単語列を関係パタンとすると,「癌を抑制するにんにく」という文から「X を抑制する Y」という関係パタンが得られるが，この関係パタンは「癌を強く抑制するリコピン」という文に適用することができない．また,「にんにくは癌の発生を抑制する」という文から名詞間の単語列を関係パタンとして抽出すると,「X は Y」という，特定の関係を表さないような表現を抽出してしまう．日本語を対象にする場合，語順の制約の緩い言語に応じた関係パタンの抽出法を考える必要がある．

また，収集した関係パタンは関係の種類毎にまとめあげておくことが望ましい．例えば,「にんにくは癌を予防する」,「リコピンは癌に効果的である」という文について，それぞれ prevent⟨ にんにく，癌 ⟩, prevent⟨ リコピン，癌 ⟩ というように，prevent 関係を述べた文であると認識するためには,「X は Y を予防する」と「X は Y に効果的である」という関係パタンが共に prevent 関係を表すことを認識しておく必要がある．Open IE では人手による事前知識を利用しないため，この関係パタンのまとめあげにはクラスタリングのような処理が必要である．あらゆる関係インスタンスを収集するために，大量の関係パタンを収集し，全パタン間の類似度を計算するには，パタン数を n とすると $O(n^2)$ の計算時間を要する．パタンの数が 10 万，100 万と大規模になると，類似度計算だけでも莫大な時間を要する．

本論文では，大規模な日本語 Web 文書から関係インスタンスと考えられる名詞対をあらかじめ簡単なヒューリスティクを用いて収集しておき，この名詞対を用いて，大量の関係パタンを収集する．収集した関係パタンについて，高速な近似類似度計算手法によって，パタン間の類似度を計算し，実用的な時間でクラスタリングを行う手法を提案する．この関係パタンの収集，およびクラスタリングによって，日本語 Web 文書中のあらゆる関係インスタンスの抽出を可能とするような，関係パタンの知識を構築する．

2 関連研究

Open IE システムには，あらかじめ関係パタンとなりうる品詞列を人手で定めたもの [4] や，小規模な関係インスタンスのデータセットから，関係に依存しないパタンの規則を獲得するものがある [9]．Fader らは，動詞や動詞句など，関係パタンとなりうる品詞列を人手で与え，関係パタンとインスタンスを認識する手法を提案した [4]．英語のように語順の制約が強い言語では品詞列は有効であるが，日本語のように語順の自由度が高い言語には適当でない．Wu らは Wikipedia のページと Infobox を利用して，関係インスタンスについて言及した文の集合を収集し，この文集合から品詞列や単語間の依存構造など，パタンの規則を獲得する手法を提案した [9]．Wu らは Wikipedia からの情報抽出の結果である DBpedia という資源を用いているが，このような資源の充実していない言語には，この手法は適用できない．

上記の研究は，関係インスタンスを列挙する事に主眼を置いており，関係の種類の特定を行っていない．大規模に収集してきた関係パタンについて，同一の関係を表すパタンをまとめあげ，関係の種類毎にインスタンスを認識できる知識として整理しておこうという研究が存在する [3, 7]．Min らは，パタンの曖昧性の解消と類似度計算の高速化のために，前処理として，各パタンと共起する関係インスタンスの名詞をクラスタリングしておく手法を提案した [7]．しかしながらこの手法は，各パタン毎に名詞をクラスタリングするアルゴリズムとなっており，全体での処理時間が名詞のクラスタリングに要する時間に応じて増大する．DeSaeger らは，大規模な日本語 Web 文書から構築した名詞クラスを関係パタンに導入し，パタン間の類似度の測定を行い，高精度で関係インスタンスを獲得可能なパタンを収集した [3]．この研究成果は関係パタンの類似度計算結果として公開されている．本研究では，大規模なパタン集合について，高

速で近似的な近傍点探索により，ある程度類似度が高いと推測されるパタンを抽出する．類似度が高いと推測されたパタン間で正確な類似度を計算し，全体の計算時間を短縮する．

3 提案手法

本節では，提案手法の概要を説明する．提案手法ではまず，コーパス中における，各名詞の頻度および共起頻度から，関係を持つと推測される名詞対を抽出する．次に，その名詞対を結ぶ表現を関係パタンとして抽出する．次に，各関係パタンについて，高速な近傍点探索手法により，一定値以上の類似度を持つと予測されるパタンを列挙する．各関係パタンについて，これら列挙したパタンとの正確な類似度を計算し，得られた類似度を元にクラスタリングを行い，関係パタンをまとめあげる．最終的に，得られた関係パタンのクラスタを，関係パタンの知識として出力する．

3.1 名詞対の抽出

多くの関係パタンの獲得が可能であるよう，コーパス中で一定値以上の出現頻度を持つ名詞を対象に，関係インスタンスと考えられる名詞対を抽出する．なお，本研究では，コーパス中での頻度上位 100 万の名詞を用いた．

「日常的な食物では，特にニンニクが癌に効果的だ」という文における ⟨ ニンニク，癌 ⟩ のように，関係インスタンスである名詞対は，同一の文に出現する事が多いと考えられる．ある名詞ペアが関係インスタンスになり得るかを判定するために，文内の共起頻度を用いるのは自然なアイディアである．

しかしながら，共起頻度のみに基づいた場合，⟨ 私，いつか ⟩ や ⟨ 自分，今日 ⟩ のように，名詞単体での出現頻度が高いため，共起頻度も高いが，関係インスタンスではないペアを抽出してしまう恐れがある．そこで，本研究では，式 (1) の値を用いて，名詞間の相関を測定し，この値の高い名詞対を抽出する．

$$\mathrm{score}(w_i, w_j) = \frac{\mathrm{cofreq}(w_i, w_j)}{\mathrm{freq}(w_i) * \mathrm{freq}(w_j)} * \mathrm{dis}(w_i, w_j) \quad (1)$$

$$\mathrm{dis}(w_i, w_j) = \frac{\mathrm{cofreq}(w_i, w_j)}{\mathrm{cofreq}(w_i, w_j) + 1} * \frac{\min(\mathrm{freq}(w_i), \mathrm{freq}(w_j))}{\min(\mathrm{freq}(w_i), \mathrm{freq}(w_j)) + 1} \quad (2)$$

ここで，w_i や w_j は名詞であり，$\mathrm{freq}(w_i)$，$\mathrm{freq}(w_j)$ はそれぞれ名詞 w_i，w_j のコーパス中での出現頻度，$\mathrm{cofreq}(w_i, w_j)$ は名詞 w_i と w_j のコーパス中での共起頻度である．すなわち，式 (1) の右辺第一項は，名詞 w_i や w_j が出現する際に，共起している回数が多いほど，大きな値となる．

しかしながら，この値は，$\mathrm{freq}(w_i)$ や $\mathrm{freq}(w_j)$ が少ない場合に，大きくなりすぎてしまう．これを防ぐために，式 (2) のような，discount factor が提案されている [8]．これは，名詞 w_i や w_j が十分な頻度で出現していないときや，$\mathrm{cofreq}(w_i, w_j)$ の値が小さいときに，$\mathrm{score}(w_i, w_j)$ の値を抑える働きがある．最終的に，式 (1) から求まる，$\mathrm{score}(w_i, w_j)$ の上位 M 個の名詞対を抽出する．なお，本研究では，$M = 100$ 万とした．

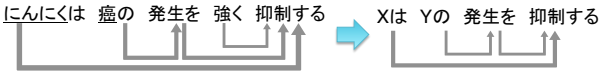

図 1: 係り受け関係に基づいた関係パタンの抽出

3.2 関係パタンの抽出

抽出された名詞対について，コーパス内の文中でその間を結ぶ表現を，関係パタンとして抽出する．1 節で述べたように，日本語は単語の順序についての制約が緩い．本研究では，係り受け関係を用いて関係パタンを表現する．具体的には，対象の名詞対を含む文節の，係り受けパスを関係パタンとして抽出する．

係り受け関係に基づいた関係パタンの例を図 1 に示す．図 1 では,「にんにくは癌の発生を強く抑制する」という文について，各文節同士の係り受け関係を，文節間に記した矢印で表している．例として，⟨ にんにく，癌 ⟩ が 3.1 節の処理で抽出された名詞対であったとすると，この文について，これらの名詞を含む文節はどちらも「抑制する」という文節にかかる．このため，係り受けパスを抽出し，抽出の根拠とした名詞部分については X，Y のように変数化することで，図 1 の下部に記したように,「X は→抑制する_Y の→発生を→抑制する」というパタンを得られる．

抽出した関係パタンの中には，コーパス中に一回しか出現しないパタンのように，インスタンスを抽出するという観点からは有用でないものが含まれている．このため，対象の名詞対との共起頻度の合計によってパタンの頻度を測定し，頻度の高いパターンのみをクラスタリングの対象とする．この関係パタンの頻度が，しきい値 α を超えるものを，クラスタリング対象のパタンとして，獲得する．

3.3 類似度計算

獲得した関係パタンについて，同一の関係を表すパタンをまとめあげることで，関係の種類毎にインスタンスを獲得できる関係パタンが得られる．関係の種類毎にパタンをまとめあげるため，パタン間の類似度を測定し，クラスタリングを行う．類似度の計算には，各パタンと共起する名詞対とその頻度を素性として用いる．

ここで，パターンベクトルの次元を k，パターンの数を n とすると，全パターンペアの類似度の計算には $O(kn^2)$ の時間を要する．これは，大規模に収集したパタン間（例えば $n = 1,000,000$）の類似度計算には，膨大な時間がかかってしまうことを意味する．一方で，同一の関係を表すパタンはせいぜい 100 のオーダー程度しか存在しないであろうという直感がある．関係パタンをまとめあげるという目的では，この同一の関係を表すパタンを絞り込めていればよく，全パタン間の正確な類似度計算は必要ない．そこで，本研究では，高速な近似近傍点探索手法により，類似度が高いと推測される関係パタンを絞り込む．その後，絞り込んだパタン間について，正確な類似度を計算し，それ以外のパタン間の類似度はゼロとして，クラスタリングを行う．

本研究では，近似近傍点探索手法として，Locality Sensitive Hashing（LSH）を用いる．LSH とは，似た素性ベクトルを持つ要素が，高い確率で同じ値を取るようなハッシュ関数を利用し，近傍点を確率的に求めるアルゴリズムである．LSH では，元の要素間に対する距離尺度

に応じたハッシュ関数を構築する必要がある．本論文では，Charikar らによって提案された，コサイン類似度に対するハッシュ関数を用いる [2]．このハッシュ関数は，元の要素の素性ベクトル $\boldsymbol{u}$ に対し，同一次元数のランダムなベクトル $\boldsymbol{r}$ を用いて，式 (3) のように定義される．

$$h_{\boldsymbol{r}}(\boldsymbol{u}) = \begin{cases} 1 & \text{if } \boldsymbol{r} \cdot \boldsymbol{u} \geq 0 \\ 0 & \text{else} \end{cases} \quad (3)$$

このハッシュ関数では，元の要素の素性ベクトル u と v 間について，

$$\Pr[h_{\boldsymbol{r}}(\boldsymbol{u}) = h_{\boldsymbol{r}}(\boldsymbol{v})] = 1 - \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{\pi} \quad (4)$$

が成り立つ．

式 3 は言い換えれば，ランダムなベクトル $\boldsymbol{r}$ によって得られる $h_{\boldsymbol{r}}(\boldsymbol{u})$ と $h_{\boldsymbol{r}}(\boldsymbol{v})$ は，$\frac{\theta(\boldsymbol{u},\boldsymbol{v})}{\pi}$ の確率で異なることを示している．$\boldsymbol{u}$ と $\boldsymbol{v}$ のコサイン類似度が $cos(\theta(\boldsymbol{u}, \boldsymbol{v}))$ であることを考えると，d 個のランダムなベクトルと式 (3) から得られる d 次元のビットベクトルについて，$\boldsymbol{u}$ と $\boldsymbol{v}$ のコサイン類似度が β であるとすると，おおよそ $d \times \frac{arccos(\beta)}{\pi}$ ビットの違いが存在する．すなわち，d 次元のビットベクトルについて，ハミング距離を測る事で，元の要素間のコサイン類似度がしきい値以上の要素対を得る事ができる．このように，LSH を用いる事で，計算対象のベクトルの次元を，元々の次元 k よりも，遥かに小さい d に削減することができる．さらに，ビットベクトル同士のハミング距離の計算は，ビットベクトル間の排他的論理和を計算し，1 が立っているビットをカウントするだけで済むため，元々の素性ベクトル間の類似度計算よりも，計算が容易である [1]．

LSH ではコサイン類似度がしきい値以上のパタン対を近似的に得る事ができる．実際に利用する際は，類似度のしきい値を実際に設定したい値よりも下げておくことにより，類似パターン対を取りこぼさないようにしておく．最終的に，LSH を用いて得られたパタン対について，正確なコサイン類似度を測定し，それ以外のパタン間の類似度はゼロとして，クラスタリングを行う．クラスタリングには，階層的クラスタリング手法である群平均法を適用し，クラスタを形成する際の類似度のしきい値を γ とした．クラスタリング後，3 個以上の要素を持つクラスタを，最終的な出力とする．

4 実験

4.1 実験設定

日本語 Web 文書約 60 億文をコーパスとして，パタンを抽出・クラスタリングする実験を行った．係り受けパタンの抽出には，日本語係り受け解析器である cabocha[2] の出力を利用した．関係パタンの頻度のしきい値は $\alpha = 300$ とした．このとき，関係パタンは 91,481 種類得られた．コサイン類似度 0.3 以上のパタン対を用いてクラスタリングを行った．

本研究で収集するパタン集合は，関係インスタンスを獲得するために用いるので，得られたパタンにより，関係インスタンスをどの程度獲得できるかで評価する事が望ましい．本実験では，Wikipedia から単純なルールで抽出できる関係インスタンスとして,「人名が作品を書いた」という著作関係,「会社（人物）が物を製作した」という製造品関係,「場所（建物）が場所に存在する」という関係を採用した．

表 1: クラスタリング結果と正解データとの比較

関係	適合率 (%)	再現率 (%)
著作	91.3	44.8
製造品	79.7	19.0
所在地	70.5	36.1

表 2: 各関係のクラスタに含まれるパタンの例

関係	関係パタン
著作	X の→作者→ Y 氏，X の→作者→ Y 先生
製造品	X の→家庭用ゲーム機→ Y，X の→ゲーム機→ Y
所在地	X の→情報ポータルサイトで→満載です _Y の→飲食店情報が→満載です

Wikipedia から抽出した正解データで評価する場合，「トヨタ」と「トヨタ自動車」のような，表記揺れを吸収する事ができない．また，今回の手法では，相関の強い名詞対 100 万対のみがインスタンス候補となっているため，この候補からもれた名詞対は抽出できない．そこで，Wikipedia から抽出したインスタンスで，かつ 100 万の名詞対に含まれるものに限定して評価したいが，正解インスタンス数が非常に少なくなってしまう．これを解消するため，システムが出力したクラスタのうち，Wikipedia から得た正解の関係インスタンスを含んでいた複数のクラスタから，ランダムに名詞対のサンプリングを行い，人手で正否を判定したうえで，正解データに加えた．最終的に，著作関係，製造品関係，所在地関係について，それぞれ 489，268，624 個の正解インスタンスを得た．

4.2 結果

しきい値 γ を 0.3 として群平均法によるクラスタリングを行った結果，4,534 個のクラスタが出力された．本手法は Open IE のため，どのクラスタがどの関係と対応づくか不明である．正解データに含まれる著作関係，製造品関係，所在地関係とクラスタを対応づけるため，各クラスタと正解データとの適合率を測定し，各関係について最も高い適合率を出したクラスタを対応づけた．このようにして選択したクラスタと正解データを比較した結果を表 1 に示す．

出力されたクラスタの適合率は非常に高く，選択されたクラスタが，別の関係にある名詞対を排除できていることが分かる．しかしながら，再現率はどの関係でも 50%を下回っており，芳しくない結果に見える．関係インスタンスの再現率を評価するため，Wikipedia から抽出した正解データとクラスタを自動的に対応づけたが，この対応付けが上手くいっていない可能性がある．また，製造品関係の再現率が非常に低い理由は，本来同種の関係である関係パタンを同一のクラスタにまとめあげる事ができなかったからである．

表 1 で評価した各関係のクラスタに属している，関係パタンの例を表 2 に示す．表 2 より，製造品関係については〈 ゲーム会社，ゲーム機 〉，所在地関係については〈 地名，詳細な地名 〉とかなり詳細な関係を表すクラスタが正解データと対応づいたと推測される．見方を変えれば，Open IE では関係の粒度を制御できないことが問

[1] 1 が立っているビットをカウントするという処理は，Intel SSE 4.2 では popcnt という専用の命令で高速に実行できる．

[2] https://code.google.com/p/cabocha

表 3: LSH を利用した計算時間

ビット数	変換（分）	ハミング距離（分）	正確な類似度（分）	合計（分）
256	12	11	241	264
512	25	8	78	111
1024	45	10	14	69
2048	86	17	4	107
元の要素	-	-	663	663

表 4: LSH で抽出したパタンペアの再現率

ビット数	0.1	0.15	0.2	0.25	0.3
256	87.8%	95.0%	98.2%	99.5%	99.9%
512	89.8%	97.0%	99.4%	99.9%	100.0%
1024	92.5%	98.9%	99.9%	100.0%	100.0%
2048	93.9%	99.6%	99.9%	100.0%	100.0%

題であり，Open IE システムの評価の難しさが窺える．一方で，同一の関係としてまとめあげるべき関係パタンが別々のクラスタになってしまっているケースも見受けられる．この問題は，関係パタン間の類似度の計算に名詞対を素性として用いているため，素性空間が疎になりやすく，似ているパタンの類似度が高くなりづらいという事が理由として上げられる．関係パタンの素性ベクトルの次元圧縮や，関係パタンを構成する内容語の情報を利用するなどして，パタンの類似度計算の精度を高める事は，今後の課題である．

4.3 LSH による類似度計算の評価

補足の実験として，LSH を用いたパタン間の類似度計算と，全パタン間の正確な類似度計算を計算時間および計算結果の面から比較する．

LSH でのビット数（d）を変化させたときの，元の素性ベクトルからビットベクトルへの変換時間，ビットベクトル上でのハミング距離の計算時間，LSH で類似すると予測されたパタン対の正確な類似度計算時間を表 3 に示す．表 3 には，元の素性ベクトルのまま全パタン間での正確な類似度計算を行った際の計算時間も示した．ビット数を変化させたときのハミング距離の計算時間は，512 ビットでは 8 分，1024 ビットでは 10 分，2048 ビットでは 17 分と，ビット数に比例して増加するが，類似度計算全体に占める割合は大きくない．

表 3 より，ビット数を増やすほど，類似すると予測されたパタン対の正確な計算時間は短くなるが，ビットベクトルに変換する時間，ビットベクトル上でハミング距離を計算する時間は増加してしまう．ビット数の決定は，類似度の近似精度と，計算時間のトレードオフがあり，今回の実験では，1024 ビットを用いたときが最も速くなった．ただし，元の素性ベクトルを用いた全パタン間の正確な類似度計算に要する時間と比べると，LSH を利用した計算時間は大幅に短縮されている．

LSH で類似パタン対を推定する場合，実際には，似ているペアが抽出されない可能性がある．コサイン類似度が 0.1 以上となるようなハミング距離を持つパタン対を LSH で抽出したとき，どの程度もれなく抽出できるかを表 4 に示した．表 4 の各列は，実際のコサイン類似度に対し，LSH でパタン対をどの程度抽出できたかの再現率を示している．表 4 より，LSH でパタンベクトルを 1024 ビットに圧縮し，コサイン類似度が 0.1 以上となるようなハミング距離を持つパタン対を抽出したとき，コサイン類似度 0.2 以上については 99.9%以上，コサイン類似度 0.25 以上については 100.0%の再現率を達成している．従って，LSH では，実際に使用する値よりも少々低めのコサイン類似度に対応するハミング距離を持つパタン対を抽出しておけば，類似度の高いパタンペアを取得もれなく獲得できると考えられる．

5 まとめ

本論文では，大規模な Web 文書から大量のパタンを抽出し，LSH という近似近傍探索手法によって，高速に類似度の計算を達成する手法を提案し，その効果を示した．さらに，類似度の計算結果を用いてパタンのクラスタリングを行い，様々な種類の関係インスタンスを獲得できるような，関係パタンの知識を構築した．

構築した関係パタンの知識は，精度は高いものの，同一の関係であるパタンが別々の種類（別のクラスタ）になってしまっている問題がある．これのマージや，関係間の関係を整理し，より広く関係インスタンスを収集できる知識を構築する事は今後の課題である．

謝辞

本研究は，JST 戦略的創造研究推進事業 CREST および，JST 戦略的創造研究推進事業「さきがけ」から部分的な支援を受けて行われた．

参考文献

[1] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the IJCAI*, pp. 2670–2676, 2007.

[2] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of STOC*, pp. 380–388, 2002.

[3] Stijn De Saeger, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, and Masaki Murata. Large scale relation acquisition using class dependent patterns. In *Proceedings of ICDM*, pp. 764–769, 2009.

[4] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the EMNLP*, pp. 1535–1545, 2011.

[5] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st ACL*, pp. 1608–1618, 2013.

[6] Shinzato Keiji, Shibata Tomohide, Kawahara Daisuke, and Kurohashi Sadao. Tsubaki: An open search engine infrastructure for developing information access methodology. 情報処理学会論文誌, Vol. 52, No. 12, 2011.

[7] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of EMNLP*, pp. 1027–1037, 2012.

[8] Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *Proceedings of NAACL*, pp. 321–328, 2004.

[9] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of ACL*, pp. 118–127, 2010.