6S-4

相互関係を利用した属性抽出手法

谷 直紀 † ボッレーガラ ダヌシカ † 石塚 満 †
† 東京大学大学院情報理工学系研究科

1 はじめに

属性とはあるものに共通して備わっている特徴のことであり，人の場合は出身地・国籍などが属性になる．これらの属性に対応する属性値 (東京・日本など) を求めることを属性抽出という．属性値は質疑応答タスクなど他のタスクにも利用できるため，属性抽出は重要である．

属性抽出について様々な研究が行われているが，ほとんどの手法は属性値を個別に抽出している．しかし，1 つのエンティティの所有している属性は独立ではなく，共起しやすい組み合わせや共起しにくい組み合わせが存在すると考えられる．そこで本稿では，属性間の相関を利用した新しい属性抽出手法を提案する．

2 関連研究

Web ページからの属性抽出に関して様々な研究が行なわれている．渡部らはリストマッチング・正規表現・固有表現抽出を利用して属性値候補を抽出した後に，人名からの距離によって属性値を選ぶ手法を提案した [1]．本稿と同様に相互相関を属性抽出に利用した例としては，Alfonseca らの研究がある [2]．Alfonseca らはエンティティ間の相関を利用し，似たエンティティは共通の属性を持つ傾向があることを用いて属性抽出の精度を向上させた．

3 提案手法

3.1 手法の概要

属性間には相関があると考えられる．例えば「1 歳の子供は政治家になれない」ことから，職業・年齢という 2 つの属性の間に相関があることが分かる．そのため，1 つのエンティティに政治家・1 歳という 2 つの属性値が共起することはできない．このような属性同士の相関を利用することで，属性抽出の精度を高めることができる．

簡単のため，A,B,C の 3 つの属性をもつエンティティを考える．これらの属性に対応する属性値 a,b,c を選択するには，それぞれの属性値候補集合 C_A,C_B,C_C から 1 つづつ属性値を選択する必要がある．また，それぞれの属性値が a,b,c となる確率 $P(A=a,B=b,C=c)$ を最大化するような属性値を選択したい．全ての属性を一度に考えると計算量が膨大になるため，確率 P を 2 つの属性間の類似度の合計で近似すると，これらの条件を満たす定式化は以下のようになる．

$$\begin{aligned} \max. \quad & \mathrm{sim}(a,b)+\mathrm{sim}(b,c)+\mathrm{sim}(a,c) \\ \text{s.t.} \quad & a \in C_A \\ & b \in C_B \\ & c \in C_C \end{aligned}$$

ここで $\mathrm{sim}(a,b)$ は属性値 a,b の類似度である．また C_A,C_B,C_C は属性 A,B,C の属性値候補集合を表す．

本稿では属性値を頂点，属性間類似度を辺の重みとするグラフを作り，この最適化問題を最大全域木 (Maximum Spanning Tree, MST) 問題として解く．最大全域木問題とは連結無向グラフが与えられたとき，グラフを構成する辺の重みの総和が最大となる全域木を求める問題である．ここで全域とは，元のグラフの部分グラフのうち頂点集合が同じグラフを指す．ただし通常の最大全域木は全ての頂点を選択するのに対して，本手法では全ての属性を選択することに注意されたい．

Attribute Extraction from the Web based on Correlation between Attributes

†Graduate School of Information Science and Technology, University of Tokyo

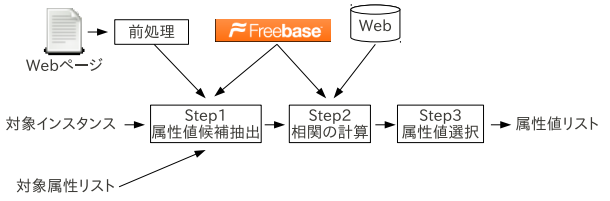

図 1: 提案手法の全体像

3.2 属性抽出の手順

提案手法の全体図を図 1 に示す．入力として Web ページ・属性リスト・人名を受け取り，3 段階の手法によって属性値リストを抽出する．Web ページには HTML タグや JavaScript など余分な情報が含まれているため，前処理として HTML タグ除去スクリプト[1] を使って除去しておく．

Step1 ではリストマッチングによって属性値候補を抽出する．まず Freebase[2]に登録されている属性値を集め，属性値候補リストを作成する．Freebase とは 1200 万件以上のエントリーによって構成されたオープンデータベースである．不特定多数の人々が自由に編集する点では Wikipedia と同じだが，Wikipedia では情報を文章で記述するのに対して，Freebase では情報は属性：属性値のペアで記述されている．今回は誕生年・出身地・国籍・職業・学位・出身大学・専攻・所属企業の 8 種類の属性を対象にする．それぞれの属性値候補の数は誕生年 1,726 個・出身地 20,165 個・国籍 345 個・職業 1,364 個・学位 125 個・出身大学 3,995 個・専攻 267 個・所属企業 10,315 個である．これらのリストに一致する単語を Web ページから見つけて，属性値候補として抽出する．

Step2 では，Step1 で抽出した属性値候補間の類似度を計算する．類似度尺度として Jaccard 係数・Dice 係数・Overlap 係数・PMI (自己相互情報量) の 4 種類を利用する．例えば Jaccard 係数は

$$\mathrm{Jaccard}(u,v)=\frac{\mathrm{hits}(u\&v)}{\mathrm{hits}(u)+\mathrm{hits}(v)-\mathrm{hits}(u\&v)}$$

となる．ただし $\mathrm{hits}(u)$ は属性値 u の検索ヒット数を，$\mathrm{hits}(u\&v)$ は属性値 u,v の AND 検索によるヒット数を表す．

検索ヒット数は Yahoo! Search BOSS[3]と Freebase を利用して取得する．Yahoo! Search BOSS を利用する場合は，属性値をクオテーションマークで囲んだクエリを作成して計算する．例えば $u=$「オバマ」,$v=$「大統領」の場合の類似度は「“オバマ”」，「“大統領”」，「“オバマ” “大統領”」という 3 種類のクエリを検索することで求められる．Freebase を利用する場合は，登録されている人物が所有する属性値を調べ，その属性値を持っている人物数を検索ヒット数とする．なお Freebase は 2010 年 11 月 16 日時点の 1,716,440 人分のデータを用いた．

Yahoo! Search BOSS と Freebase の 2 種類の検索ヒット数から類似度を計算し，組み合わせることで最終的な類似度を計算する．u,v という 2 つの属性値の類似度 $\mathrm{sim}(u,v)$ は

$$\mathrm{sim}(u,v)=\alpha\times\mathrm{webSim}(u,v)+(1-\alpha)\times\mathrm{freebaseSim}(u,v)$$

[1]http://www.oluyede.org/files/htmlstripper.py
[2]http://www.freebase.com/
[3]http://developer.yahoo.com/search/boss/

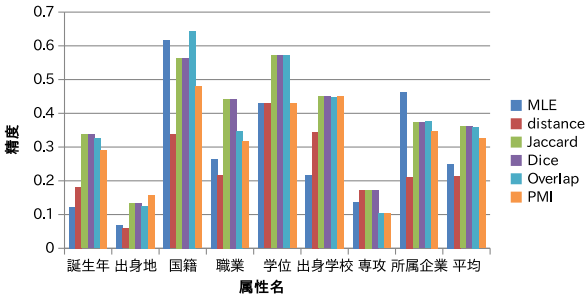

図 2: 属性抽出の精度

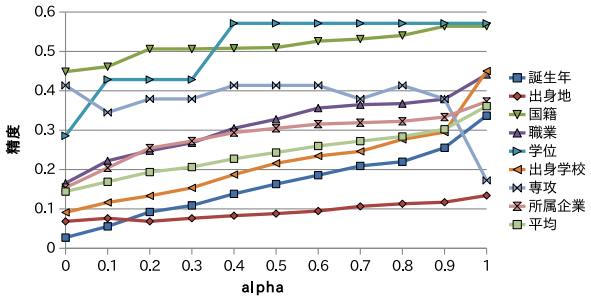

図 3: Web 類似度と Freebase 類似度の割合

と表せる．α は Web 類似度と Freebase 類似度の割合を調整するためのパラメータであり，0 から 1 の値をとる．α が大きくなるほど Freebase 類似度の影響が大きくなる．また，webSim(A,B) は Yahoo! Search BOSS の検索ヒット数を元にした類似度であり，freebaseSim(A,B) は Freebase の検索ヒット数を元にした類似度である．

Step3 では属性値候補のランク付けを行う．まず Step2 で調べた属性値間の類似度を元にグラフを作成する．グラフの頂点は属性値であり，辺には属性値間の類似度で重みをつける．このグラフの最大全域木を求めることで，全ての属性を選択しながら類似度の合計が最も大きくなる属性値リストを取得する．

最大全域木を求める手法としてプリム法 [3] を使用する．ただし同じ属性に属する属性値を複数選択しないようにすでに選ばれた属性のリストを持っておき，その属性リストに一致する場合はその属性値は選ばないようにする．

4 評価実験

Web 上のテキストから属性抽出を行い，属性の相互相関を使うことによってどの程度精度が向上するかを調べる．

4.1 実験方法

対象人物に関係した Web ページから属性抽出を行い，抽出精度の測定を行う．Freebase に対象属性が 5 種類以上登録されている人物をランダムに 100 人選択し，対象人物とする．この人名をクエリとして Web 検索を行い，検索上位 50 件のページをデータセットとする．ただし PDF など HTML 以外のページは検索結果に含めない．データ数は 100 (人) × 50 (ページ) = 5000 (ページ) となる．

それぞれの文章に対して属性抽出を行い，提案手法とベースライン手法で属性値を選択する．選択された属性値が Freebase に登録されている正解属性値の 1 つと一致した場合，正解とする．ただし正解属性値が文章中に含まれていない場合は属性抽出が不可能であるため，精度の計算には含めない．この作業を 5000 ページに対して行い，抽出精度のミクロ平均をその手法の評価とする．

ベースライン手法として MLE と Distance の 2 つを用意した．MLE では対象文章中の属性値候補の出現回数をカウントし，最も多く出現した属性値を選択する最尤推定 (Maximum Likelihood Estimation) を行う．Distance では人名からの距離を利用する．文章中に出現する対象人名にタグをつけておき，属性値候補のうち最も人名の近くに出現する属性値を選択する．人名にタグをつける際には名字・名前の組み合わせや Mr., Mrs., Miss. といった冠詞を利用して約 30 種類のバリエーションを作成し，表記揺れに対応した [1]．

また，Jaccard 係数を利用した場合に α を 0.0 から 1.0 まで 0.1 刻みで変化させ，Web 類似度と Freebase 類似度の割合による抽出精度の変化を測定した．

4.2 実験結果

属性抽出の精度を図 2 に示す．提案手法の中では，Jaccard 係数を使った場合の精度が 36%と最も高くなっている．t 検定を行なったところ，提案手法とベースライン手法の精度には有意水準 1%で有意差があった．

Jaccard 係数と Dice 係数では精度が全く同じになっている．これは $Jaccard = Dice/(2 - Dice)$ という関係が成り立つため，絶対値が異なるものの同じランキングが得られたことが原因と考えられる．

国籍・所属企業という 2 種類の属性では提案手法がベースライン手法 (MLE) に精度で劣っていた．この原因としてはデータが偏っていたことが考えられる．Freebase にはアメリカ国籍の人物が最も多く登録されているため，実験用データでも 60%の人がアメリカ国籍になっている．また，英文中に最も出現しやすい国名はアメリカであると考えられる．そのため最尤推定が最も良い精度を出したと思われる．

α を変化させた結果を図 3 に示す．$\alpha = 1.0$，つまり Freebase 類似度のみを使った場合に精度が最も高くなっている．この原因としては Web 類似度のノイズが挙げられる．Web の場合は属性値候補が出現しても正しい属性値ではない場合や，複数の人物の属性値が同一文章中に出現するなど様々なノイズが存在する．そのため Freebase 類似度に比べて精度が下がっていると考えられる．

5 おわりに

インターネットから属性を抽出する手法についてはこれまで様々な研究が行われてきたが，抽出した属性候補を選択する手法に特化した研究はあまり行われていない．また，属性同士の関係もあまり利用されてこなかった．本稿では，属性同士の相関を利用したインターネットからの属性抽出手法を提案した．提案手法は既存の属性抽出手法と組み合わせて使うことができ，応用範囲が広い．また，Web ページからの属性抽出について評価実験を行い，ベースライン手法よりも精度を向上できることを示した．

今後の課題としては，相関のある属性のみを利用して属性抽出を行なうことがある．現状では全ての属性間に相関があると仮定して属性値を選択しているが，属性の組み合わせによって相関が強いものや弱いものがあるはずである．そこで機械学習を利用してどの属性間に相関があるかを調べることで，より精度を向上させることができるだろう．

参考文献

[1] K. Watanabe, D. Bollegala, Y. Matsuo, and M. Ishizuka. A two-step approach to extracting atrtibutes for people on the web. In *Proc. Second Web People Search Task (WePS2) Workshop*, 2009.

[2] A. Enrique, P. Marius, and R. Enrique. Acquisition of instance attributes via labeled and related instances. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pp. 58–65, New York, NY, USA, 2010. ACM.

[3] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technology Journal*, Vol. 36, pp. 1389–1401, 1957.