E-008

Webにおける単語出現分布情報を用いた名詞のカテゴリ推定

Noun categorization with the word frequency distribution on the Web

宮村 祐一† 清水 勇詞† 鈴木 優†
Yuichi Miyamura Yuji Simizu Masaru Suzuki

概要

本論文では、大量の Web 文書の情報を利用して名詞のカテゴリ（地名、人名など）を推定する手法を提案する。従来手法では、推定対象の単語の近傍情報（前後数単語の情報）を用いて推定を行う。それに加え、提案手法では、同じカテゴリに含まれる単語はインターネット上の同じドメインに属する Web 文書に出現しやすいとの仮説に基づき、単語が出現するドメインの分布情報を用いることでカテゴリ推定精度の向上を試みた。Web 検索エンジンを利用した実験を行い、名詞のカテゴリ推定精度が向上することを確認した。

1. はじめに

形態素解析や固有表現抽出などの言語処理では、処理の際に辞書を情報源として利用することが多い。これらの処理では辞書の大きさが処理性能に影響を与えるため、より大規模な辞書が必要とされている。しかし、人手による辞書構築には多大な作業コストを要することから、大規模な辞書の構築は困難であった。そこで、近年、自動で辞書構築を行う研究[1, 2, 3]が盛んに行われている。

辞書に登録される単語は、その辞書の用途によって様々であるが、多くの場合に名詞が登録単語の大半を占める。これは、形態素解析用辞書のように様々な品詞を登録単語にもつ辞書においても同様である。よって、登録が必要な名詞を適切に扱うことが辞書構築では重要である。

名詞は、その単語がもつ意味によって複数のカテゴリに分類することが出来る。例えば、京大コーパス[4]の品詞体系では、名詞は地名や人名などのカテゴリに細分類されている。こうしたカテゴリ情報は固有表現抽出などの際に有用であるため、自動推定できることが望ましい。そこで、本論文では名詞のカテゴリを自動推定する手法を提案する。

なお、本論文で提案するカテゴリ推定手法は、主に辞書構築時の利用を想定しているが、以下の条件が満たされれば、辞書構築以外の場合にも利用可能である。

(ア) カテゴリ推定の対象とする単語は常に名詞とする。
(イ) 各カテゴリに属する単語が予め与えられているものとする。以降、それらをカテゴリ辞書と呼ぶ。ただし、カテゴリ辞書には推定対象単語は含まれていないものとする。
(ウ) Web もしくは Web の情報を予め取得してあるデータベースにアクセス可能な環境下で推定が行われるものとする。

2. 関連研究

辞書構築では、大まかに分けて、以下の 2 処理を行う必要がある。

I. 辞書登録が必要な単語を抽出する。
II. 抽出された単語に必要な情報を付与する。

登録が必要な単語の抽出には、研究[1]で用いられるブートストラップ手法や、研究[2]で用いられる未知語を抽出する手法などが挙げられる。抽出された単語は、必要に応じて品詞やカテゴリ情報などが付与され、辞書に登録される。

本論文では、辞書構築時に付与する情報のうち、人名や地名などの名詞カテゴリに焦点を当てるが、これらは品詞の細分類として扱われることが多い。その場合、カテゴリは品詞の一部であるため、カテゴリの付与に専用処理が用いられることは少なく、品詞推定手法によって付与されることになる。

品詞推定の従来研究として中川らの研究[5]や中川らの研究[6]が挙げられる。

中川らの研究[5]では、品詞推定対象となる単語の前後数単語の情報（以降、近傍情報とする）を用いてSVMによる学習・推定を行う。この手法は、推定対象単語の近傍に各品詞を指し示す明示的な手がかりが存在する場合、高い推定性能を持つ。例えば、推定対象単語の後ろに「さん」や「氏」などが続く場合、その単語は人名である可能性が高いといえる。しかし、反対にそうした手がかりが存在しない場合には人名とその他の名詞を区別することは難しい。

中川らの研究[6]では、研究[5]で用いた単語近傍情報に加え、大域的な情報を利用した手法が提案された。ここで言う大域的な情報とは、ある単語の品詞推定を行う際に、その単語と同じ見出しを持つ別の単語が同一文書中でどのような品詞として用いられているかを表した情報である。例えば、「川崎」という単語はその前後に明示的な手がかりが存在しない場合、地名か人名かそれ以外の名詞か判断することが難しい。しかし、別の箇所で「川崎さん」という文字列が存在した場合、後者の「川崎」は人名である可能性が高い。そこで、後者の「川崎」が人名らしいという情報を前者の「川崎」の推定にも利用する。

これらの従来研究では、単語近傍に手がかりが現れやすいカテゴリと、そうでないカテゴリとで推定性能に大きな差が生じる。そのため、カテゴリの違いによる推定性能のばらつきが問題となる。この問題に対処するためには、単語近傍情報とは異なる何らかの情報を用いることが必要であると考えられる。

† （株）東芝　研究開発センター
知識メディアラボラトリー

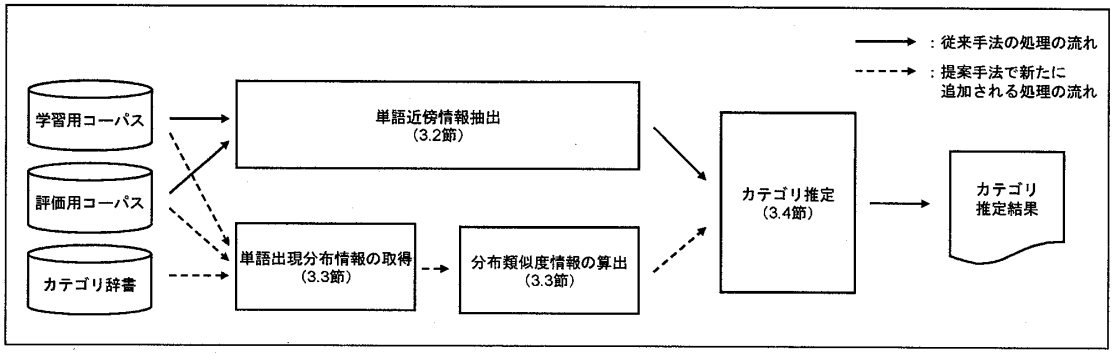

図 1　名詞カテゴリ推定の流れ

3.　提案手法

3.1　手法の概要

本論文では、以下の仮説に基づき、Web における推定対象単語の単語出現分布情報（Web 上の各ドメインにおける単語出現頻度）を用いることでカテゴリ推定性能の向上を試みる。

仮説：単語を意味によって分類した場合、同じカテゴリに含まれる単語は同じドメインに属する Web 文書に出現しやすい。

本論文で提案する推定手法の流れを図 1に示す。提案手法は以下の 4 処理からなる。

i.　単語近傍情報の抽出
ii.　単語出現分布情報の取得
iii.　分布類似度情報の算出
iv.　近傍情報と分布類似度情報を用いたカテゴリ推定

まず始めに処理 i と ii を行い、カテゴリ推定対象単語の単語近傍情報と単語出現分布情報を取得する。取得した単語出現分布情報を基に処理 iii において分布類似度情報を算出する。そして最後に処理 iv において処理 i と iii の結果を基にカテゴリ推定用の素性を作成し、それらの素性を用いてカテゴリ推定を行う。

処理 i については 3.2 節で、処理 ii と iii について 3.3 節で、処理 iv については 3.4 節で述べる。

3.2　単語近傍情報の抽出

本処理では、推定対象単語の近傍情報の抽出を行う。抽出する情報は以下の通りである。

1.　推定対象単語の前後 2 単語の文字列と品詞（4 次元）
2.　推定対象単語の文字種（1 次元）
3.　推定対象単語の 2 文字までの語頭・語尾（4 次元）
4.　推定対象単語の文字列長（1 次元）
　文字列長を 1、2、3～4、5 以上の 4 つに分類し、どの範囲に属するかを示すラベルを付与する。

抽出する単語近傍情報は従来研究[5]を参考に選定した。従来研究[5]は英語テキストを対象としたものであるため、研究[6]を参考にして抽出する語頭・語尾の長さを日本語用に変更した。情報 1～4 までを抽出することにより、1 単語に対して 10 次元の単語近傍情報が生成される。

3.3　出現分布情報の取得と分布類似度情報の算出

単語出現分布情報の取得と、取得した情報を基に分布類似度情報を算出する。分布類似度情報とは、ある単語の Web 上での出現分布がどのカテゴリの分布に近いかを示す情報である。

本処理は以下のステップによって求める。

1.　単語出現分布情報の取得
　カテゴリ辞書中の各カテゴリに属する単語が Web 上のどのドメインで何回出現しているかを調べ、単語出現分布ベクトルを作成する。単語 W の単語出現分布ベクトル V(w)は以下の式で表せる。なお、F(w,i)はドメイン i 中での単語 W の出現頻度とする。

$$V(w) = \{ F(w,1), \cdots, F(w,i), \cdots \} \quad (1)$$

2.　分布類似度情報の算出

A)　作成された単語出現分布ベクトルと、その単語が属するカテゴリ名の対を用いて、カテゴリ学習を行う。学習にはSVM-multiclass[7]を用いる。

B)　学習・評価コーパスの各単語に対しても単語出現分布ベクトルを作成し、ステップ A で学習したモデルを用いて、カテゴリ推定を行う。

C)　推定結果から分布類似度情報を作成する。分布類似度情報は下記の情報で構成される。

i.　SVM 判別スコア最大となるカテゴリ名（1 次元）
ii.　2 番目に大きな SVM 判別スコアとなるカテゴリ名（1 次元）
iii.　各カテゴリに対する SVM 判別スコア（k 次元、k=カテゴリの種類数）

判別スコアを-100、-10、0、10、100 を境に 6 つの範囲に分け、属する範囲を示すラベルを付与する。

3.4　近傍情報と分布類似度情報を用いた推定

3.2 節で生成された単語近傍情報と、3.3 節で生成された分布類似度情報を用いてSVMによるカテゴリ学習・推定を行う。本処理で用いるSVMは、学習素性として文字列を扱う必要があるため、文字列の使用が可能なYamCha[8]を用いる。

本処理のカテゴリ推定結果が最終的な提案手法の結果となる。

表 1　学習・評価コーパスの構成

	人名	地名	組織名	その他
学習コーパス	4244 語	5256 語	1604 語	70916 語
評価コーパス	2782 語	1130 語	555 語	10705 語

表 2　単語出現分布情報による推定性能変化

	再現率	精度	F 値
従来手法	0.6416	0.8312	0.7242
提案 手法	0.7208	0.8193	0.7669

4. 評価実験

4.1 コーパスとカテゴリ辞書

評価実験には、人手で品詞情報が付与されている京大コーパスを用いた。京大コーパスでは、名詞を普通名詞、サ変名詞、地名など計 10 種類に細分類している。本実験では、この 10 種類の細分類のうち、意味的にまとめられている地名、人名、組織名を推定対象カテゴリとした。

学習コーパスと評価コーパスは京大コーパスを 2 分割することで作成した。2 分割された一方を学習コーパスとし、学習コーパス中に含まれる全ての名詞を用いて SVM 学習を行った。また、もう一方を評価コーパスとし、学習コーパスに含まれない名詞を 評価対象とした。学習・評価コーパスの構成を表 1 に示す。なお、評価は「その他」を除くカテゴリの再現率・精度・F 値で行う。

カテゴリ辞書には、予め用意した地名・人名・組織名・その他の名詞を用いた。用意した各カテゴリの単語の中で、学習・評価コーパスに出現しない単語を 1 万単語ずつランダムに選択し、計 4 万単語からなるカテゴリ辞書を作成した。

4.2 実験 1：単語出現分布情報の性能改善効果

4.2.1 検索エンジンを用いた出現分布情報取得

Web における単語出現分布情報を取得するには、Web 上の全テキストを収集し、それらの中で当該単語がどのドメインに何回出現するかを集計する必要がある。しかし、この集計は処理量が非常に膨大となる。そこで、本実験では単語出現分布情報を取得する際に、検索エンジンによる検索結果順位に閾値を設け、閾値以上の検索結果のみを用いることで、処理量を低減させる。

検索エンジンを用いた単語出現分布情報取得手順は以下の通りである。

1. 検索エンジンを用いて単語Wを検索し、検索結果を取得する。検索エンジンはYahoo[9]を用いる。
2. 検索結果の上位 100 件の URL を取得する。
3. 取得した URL のドメイン名(厳密にはサーバ名)の箇所を抽出する。
   例：抽出前 www.toshiba.co.jp/rdc/index_j.htm
   抽出後 www.toshiba.co.jp/
4. 取得した各ドメイン名の数を集計する。

4.2.2 実験結果

実験には以下の 2 手法を用いた。

従来手法：3.2 節の近傍情報を素性とし SVM で学習、推定を行ったもの。

提案手法：3.2 節の近傍情報に加え、検索エンジンによって取得した単語出現分布情報を用いて SVM の学習、推定を行ったもの。

表 3　カテゴリごとの性能変化（F 値）

	人名	地名	組織名
従来手法	0.8318	0.6137	0.3380
提案 手法	0.8467	0.7273	0.4175

実験結果を表 2、3 に示す。表 2 は人名、地名、組織名を合わせた推定結果を表す。また、表 3 はカテゴリごとの推定結果を表す。

表 2 から、単語出現分布情報を用いることで再現率が改善することが分かった。F 値としても改善していることから、単語分布情報による推定精度改善効果が確認された。これは、従来手法では単語近傍に手がかりが乏しいために検出できなかった単語が、提案手法では単語出現分布情報を用いることで検出できるようになったためと考えられる。

また、表 3 から、従来手法で推定性能が高い人名に対しては、単語出現分布情報による性能改善効果は小さいことが分かる。しかし、推定性能が低い地名や組織名に対しては大幅に性能改善していることが分かる。これにより、単語出現分布情報を用いることで、カテゴリの違いによる性能のばらつきが低減することを確認した。

ただし、ばらつきは低減したものの人名と組織名では依然大きな性能差があり、さらなる改善が必要である。

4.3 実験 2：検索エンジンによる影響検証

4.2 節の実験では、検索エンジンを用いることによって単語出現分布情報の取得に要する処理量を大幅に低減させることができた。しかしその反面、次の問題点が考えられる。

問題点：検索エンジンは何らかのアルゴリズムに従って検索結果を順位付けしている。そのため、検索結果の上位だけを用いて単語出現分布情報を収集した場合、実験結果のカテゴリ推定性能が本来の性能と異なる可能性がある。

そこで本実験では、複数の検索エンジンで実験を行い、かつ、取得する URL の件数を変化させたときの推定性能を比較することで、検索エンジンによる影響を確認する。

実験条件は以下の通りである。

- 検索エンジン：Yahoo、Live Search[10]
- URL 取得件数：0 ～ 500

実験結果を図 2に示す。実験結果より、取得する URL 件数が少ない(50 件)場合には、検索エンジンの違いによって改善効果に差が見られた。しかし、取得件数を増加させることにより、どちらの検索エンジンでも F 値が 0.77 辺りに収束していることが分かった。この結果から、取得する URL 件数を十分に大きな値(100 件以上)とすることで、検索エンジンの違いによる影響は限定的であると考えられる。

また、取得する URL 件数を増加させることで推定性能が F 値 77%付近に収束しているため、Web 上の全データで

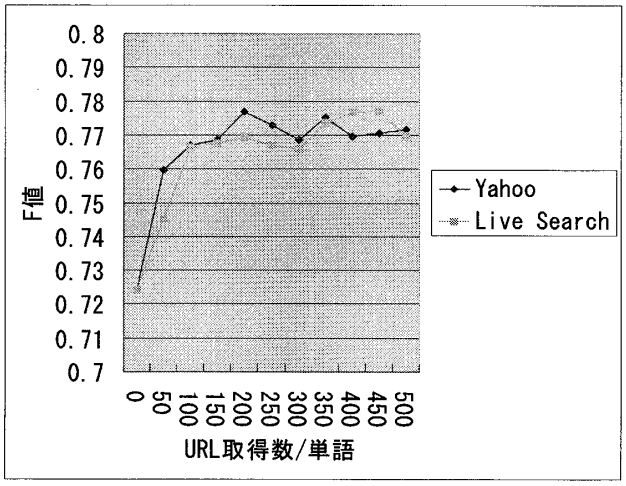

図 2　URL 取得数による推定性能の変化

実験した場合でも同程度の性能と考えられる。ただし、実験で取得した URL 件数が十分に大きいとは言えないため、本実験の結果に検索エンジンによる影響がまったく含まれていないとは言い切れない。

以上のことから、現状確認しうる範囲内では 3.1 節の仮説による推定性能改善があったと考えられる。しかし、検索エンジンによる影響が含まれている可能性があるため、URL 取得件数をさらに増加させる等の方法によって、更なる検証が必要である。

5. おわりに

本論文では、人名や地名などの名詞カテゴリを自動推定する手法を提案した。従来手法の問題点として挙げられる“カテゴリの違いによる推定性能のばらつき”を低減させるため、単語出現分布情報を用いたところ、ばらつきが低減されることを確認した。本実験では、単語出現分布情報の取得に要する処理量を減らすため、検索エンジンを用いて取得を行った。そのため、実験結果に検索エンジンによる影響が含まれる懸念が生じた。そこで、複数の検索エンジンを用いて実験を行ったところ、確認した範囲内では検索エンジンによる影響は限定的であった。以上のことから、単語出現分布情報にカテゴリ推定性能改善効果があったと考えられる。

今後の課題として、以下の点が挙げられる。

(1)　推定性能ばらつきの更なる低減方法の検討
(2)　カテゴリの種類数を増加させた場合の性能検証
(3)　単語出現分布情報の取得方法の検討
　(3.1) 同表記異義語を考慮した方法
　(3.2) 検索エンジンの影響を受けない方法
　(3.3) Web 以外の大規模テキストデータからの取得

1 つ目の課題として、カテゴリの違いによる推定性能ばらつきの更なる低減が挙げられる。提案手法で用いた単語出現分布情報により、性能ばらつきの低減が確認できたものの、依然として大きなばらつきが存在する。よって、更なる性能ばらつき低減が課題となる。

2 つ目の課題として、「食べ物」などのように、従来、推定対象として用いられることが少ないカテゴリの推定性能検証が挙げられる。本実験では、人名や地名などのカテゴリを推定対象としたが、これらのカテゴリは多くの実験で用いられてきた。そこで、対象とされることが少ない「食べ物」などのカテゴリにおいても同様の結果が得られるか確認することが重要である。

3 つ目の課題として、単語出現分布情報の取得方法の検討が挙げられる。本実験で行った検索エンジンによる取得方法では、同表記異義語を区別することが出来ないことや、検索エンジンによる推定性能への影響といった問題がある。よって、こうした問題を解決する取得方法を検討する必要がある。また、単語出現分布情報の情報源として、本実験では Web を用いたが、Web 以外の大規模テキストデータを利用した実験も考えられる。

これらの課題を引き続き検討していく必要がある。

参考文献

[1] 水口弘紀, 河合英紀, 土田正明, 久寿居大, “Web 知識を利用したブートストラップによる辞書増殖手法”, 第 18 回データ工学ワークショップ(DEWS2007), 2007.
[2] 柳原正, 池田和史, 松本一則, 滝嶋康弘, “情報量基準に基づく形態素解析用辞書の自動獲得方式”, 第 8 回情報科学技術フォーラム(FIT2009), 2009.
[3] 福島健一, 鍜治伸裕, 喜連川優, “機械学習を用いたカタカナ用言の獲得”, 言語処理学会第１３回年次大会, 2006.
[4] 京都大学テキストコーパス Version 4.0. http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/corpus.html
[5] 中川哲治, 工藤拓, 松本裕治, “Support Vector Machine を用いた未知語の品詞推定”, 情報処理学会研究報告, 2001.
[6] 中川哲治, 松本裕治, “大域的な情報を用いた未知語の品詞推定”, 情報処理学会論文誌, 2008.
[7] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables", Journal of Machine Learning Research (JMLR), 6(Sep):1453-1484, 2005. http://jmlr.csail.mit.edu/papers/volume6/tsochantaridis05a/tsochantaridis05a.pdf
[8] YamCha, http://chasen.org/~taku/software/yamcha/
[9] Yahoo! API, http://developer.yahoo.co.jp/
[10] Live Search, http://www.live.com/