LE_005

受け身文の能動文への変換における機械学習を用いた格助詞の変換に関する実験

Experiment on transformation of Japanese case particles based on machine learning when transforming passive sentences into active sentences

村田 真樹† Murata Masaki　金丸 敏幸†‡ Toshiyuki Kanamaru　白土 保† Tamotsu Shirado　井佐原 均† Hitoshi Isahara

1．はじめに

本研究の目的は，日本語の受け身文を能動文に変換する際に変更されるべき格助詞を機械学習を用いて自動変換することである．日本語の受け身文の例を図 1 にあげる．図 1 の文の日本語の接尾辞「れた」は受動態を示す助動詞であり，この文は受け身文である．この文に対応する能動文を図 2 に示す．図 2 の文が能動文に変換されるときは，(i)格助詞「に」は格助詞「が」に，(ii)格助詞「が」は格助詞「を」に変換される．本研究では，この格助詞の変換(例：格助詞「に」の格助詞「が」への変換)を，研究の対象とする．

犬に私が噛まれた．
図 1　受け身文

犬が私を噛んだ．
図 2　能動文

受け身文の能動文への変換は，文生成，言い換え[1]，文の平易化／言語運用支援[2]，自然言語文からの知識獲得や情報抽出，質問応答システム[3]と多くの研究分野で役に立つものである．例えば，質問応答システムでは，質問文が能動文，答えが受動文で書かれている場合，質問文と答えを含む文で，文の構造が異なるために，質問の答えを取り出すのが困難な場合がある．このような問題も受け身文の能動文への変換ができるようになると解決する．このように受け身文の能動文への変換は，自然言語処理において重要である．

受け身文の能動文への変換における格助詞の変換は，変換される助詞が動詞やその使われ方に依存して変わるので，簡単な問題ではない．従来手法[4,5,6]では，この問題はどのように格助詞を変換すればよいかを記載した格フレーム辞書を用いて対処されていたが，辞書にすべての動詞とその動詞の使い方を書くのは困難なため，この格フレーム辞書を用いる方法は不十分である．これに対して，われわれは教師ありデータを使った機械学習手法に基づいて格助詞の変換を行なう．我々は文献[7]において，同様に機械学習手法に基づいた格助詞の変換の研究を行ったが，本研究ではこの研究に比べて機械学習で用いる素性を充実させることで 3.3%(89.09%から 92.39%)の精度向上を実現している．また，文献[7]では，従来手法との比較実験が不十分であった．本研究では従来手法との比較実験も行い，従来手法の精度が 80%弱であり，本手法が従来手法に比べて高い性能であることを確認している．

† 独立行政法人 情報通信研究機構
{murata,kanamaru,shirado,isahara}@nict.go.jp
National Institute of Information and Communications Technology.

‡ 京都大学
kanamaru@hi.h.kyoto-u.ac.jp
Kyoto University.

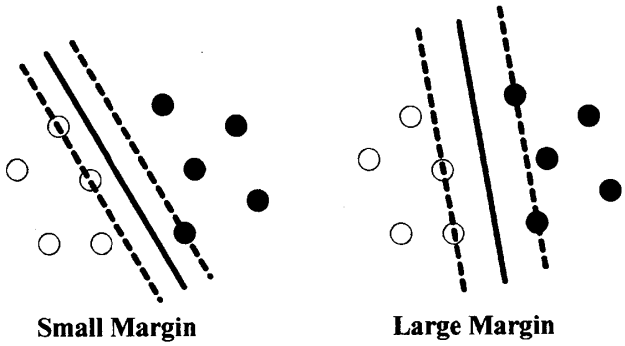

図 3　マージン最大化

2. 機械学習手法(サポートベクトルマシン法)

本研究では機械学習手法としてはサポートベクトルマシン法を用いた．これは，サポートベクトルマシン法が多くの研究分野[8,9,10]において他の手法に比べて比較的よい成績をおさめているためである．本研究ではサポートベクトルマシン法を用いるが，この研究のために他の機械学習手法を用いることもできる．本節ではわれわれが使ったサポートベクトルマシン法について説明する．

サポートベクトルマシン法は，空間を超平面で分割することにより 2 つの分類からなるデータを分類する手法である．このとき，2 つの分類が正例と負例からなるものとすると，学習データにおける正例と負例の間隔(マージン)が大きいもの(図 3 参照．図の白丸，黒丸は，正例，負例を意味し，実線は空間を分割する超平面を意味し，破線はマージン領域の境界を表す面を意味する．)ほどオープンデータで誤った分類をする可能性が低いと考えられ，このマージンを最大にする超平面を求めそれを用いて分類を行なう．基本的には上記のとおりであるが，通常，学習データにおいてマージンの内部領域に少数の事例が含まれてもよいとする手法の拡張や，超平面の線形の部分を非線型にする拡張(カーネル関数の導入)がなされたものが用いられる．この拡張された方法は，以下の識別関数を用いて分類することと等価であり，その識別関数の出力値が正か負かによって二つの分類を判別することができる[11,12].

$$f(\mathrm{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathrm{x}_i, \mathrm{x}) + b\right) \quad (1)$$

$$b = -\frac{max_{i,y_i=-1} b_i + min_{i,y_i=1} b_i}{2}$$

ただし，x は識別したい事例の文脈(素性の集合)を，x_i と $y_i\,(i=1,...,l, y_i \in \{1,-1\})$は学習データの文脈と分類先を意味し，関数 sgn は，

$$sgn(x) = \begin{array}{ll} 1 & (x \geq 0) \\ -1 & (otherwise) \end{array} \quad (2)$$

であり，また，各 α_i は式(4)と式(5)の制約のもと式(3)の $L(\alpha)$ を最大にする場合のものである．

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i y_j K(\mathrm{x}_i, \mathrm{x}_j) \quad (3)$$

$$0 \leq \alpha_i \leq C(i=1,...,l) \quad (4)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (5)$$

また，関数 K はカーネル関数と呼ばれ，様々なものが用いられるが本稿では以下の多項式のものを用いる．

$$K(\mathrm{x}, \mathrm{y}) = (\mathrm{x} \cdot \mathrm{y} + 1)^d \quad (6)$$

C，d は実験的に設定される定数である．本稿ではすべての実験を通して C，d は 1 と 2 に固定した．ここで，$\alpha_i \geq 0$ となる x は，サポートベクトルと呼ばれ，通常，式(1)の和をとっている部分はこの事例のみを用いて計算される．つまり，実際の解析には学習データのうちサポートベクトルと呼ばれる事例のみしか用いられない．

サポートベクトルマシン法は分類の数が 2 個のデータを扱うもので，通常これにペアワイズ手法を組み合わせて用いることで，分類の数が 3 個以上のデータを扱うことになる[13].

ペアワイズ手法とは，N 個の分類を持つデータの場合，異なる二つの分類先のあらゆるペア(N(N-1)/2 個)を作り，各ペアごとにどちらがよいかを 2 値分類器(ここではサポートベクトルマシン法)で求め，最終的に N(N-1)/2 個の 2 値分類器の分類先の多数決により，分類先を求める方法である．

本稿のサポートベクトルマシン法は，上記のようにサポートベクトルマシン法とペアワイズ手法を組み合わせることによって実現される．

3. 素性(機械学習による分類に用いる情報)

本研究で利用した素性を表 1 に示す．ただし，解析対象の格助詞に前接する体言を n，n の係る用言を v としている．体言 n，用言 v，品詞や統語構造の情報の特定には KNP[14]を利用した．

F14 や F15 は，他手法の解析結果を素性として追加したものである．このような手法は“スタッキング”と呼ばれ，複数のシステムの解析結果の融合に用いられている[15]．また，F16 から F21 は KNP での変換の際に用いた情報を，F22 から F26 は近藤法での変換の際に用いた情報を素性として利用している．近藤法では，格変換の際に必要な情報を登録している動詞辞書を用いる．表中の VDIC は，その変換用の動詞辞書を指す．

表1　本研究で用いた素性

F1	用言 v の品詞
F2	用言 v の単語の基本形
F3	用言 v の全単語
F4	用言 v の単語の分類語彙表の分類番号の 1,2,3,4,5,7 桁までの数字.
F5	用言 v につく助動詞列
F6	体言 n の単語
F7	体言 n の単語の分類語彙表の分類番号の 1,2,3,4,5,7 桁までの数字.
F8	用言 v にかかる体言 n 以外の体言の単語列. ただし，どういった格でかかっているかの情報を AND でつける.
F9	用言 v にかかる体言 n 以外の体言の単語集合の分類語彙表の分類番号の 1,2,3,4,5,7 桁までの数字. また，どういった格でかかっているかの情報を AND でつける.
F10	用言 v にかかる体言 n 以外の体言がとっている格
F11	同一文に共起する語
F12	同一文に共起する語の分類語彙表の分類番号の 3,5 桁の数字.
F13	変換前の格助詞
F14	KNP[14]による変換結果
F15	近藤手法[6]による変換結果
F16	IPAL にある用言 v が持つ格助詞の出現順
F17	IPAL にある用言 v の持つ述素全て
F18	IPAL にある用言 v の持つ述素
F19	体言 n の IPAL による意味素全て
F20	体言 n の IPAL による意味素
F21	用言 v が IPAL に存在するかどうか
F22	用言 v の受動態が可能かどうか(近藤手法で使用されている辞書 VDIC を参考)
F23	用言 v の必須格(VDIC による定義)
F24	用言 v の種類(VDIC による定義)
F25	近藤手法で変換の際に用いた格変換規則
F26	用言 v が近藤手法の辞書(VDIC)に存在するかどうか
F27	用言 v にかかる格助詞を持つ体言節の格助詞の出現順
F28	用言 v にかかる格助詞を持つ体言節の連続する格助詞のペア
F29	体言 n の前方に出現する全ての格助詞
F30	体言 n の後方に出現する全ての格助詞
F31	体言 n の直前に出現する格助詞
F32	体言 n の直後に出現する格助詞

4. 実験

本稿の手法を確認する実験を行った．実験には，京大コーパスにあった受け身文の 1877 個の格助詞を利用した(KNP が正しく係り受け解析できたもの)．変換後の格助詞の出現率を表 2に示す．評価には，10 分割のクロスバリデーションを利用した．機械学習の際は，複数の格助詞が正

解になる場合は，その複数の格助詞の組を正解として扱って学習を行った．

実験結果を表 3と表 4 に示す．ベースライン１の方法は変換前の格助詞を答えに出力する方法でベースライン２の方法は最も頻度の多い変換後の格助詞(すなわち，「を」)を答えとして出力する方法である．KNP は KNP の出力の結果の精度を，近藤法は近藤らの方法による結果の精度を意味する．KNP と近藤法は IPAL の辞書と VDIC の辞書に解析対象の動詞が含まれている時のみ答えを出力する．「KNP/近藤法+ベースライン X」は，IPAL の辞書や VDIC の辞書に解析対象の動詞が含まれておらず KNP/近藤法で答えを出力できなかったときにはベースライン X の方法で答えを出力する方法を意味する．村田法は先行文献[7]の方法を意味し，本稿での F1, F2, F5, F6, F7, F10, F13 のみの素性を利用する方法を意味する．評価には評価 A と評価 B の二つの評価基準を用いた．この基準は複数の格助詞が解となる場合のためのものであり，例えば評価 A は，「が」と「で」が正解の場合，出力が「が」と「で」の組の場合のみ正解とするもので，評価 B は，出力が「が」か「で」か「が」と「で」の組の場合正解とするものである．表 3は全データを利用した場合の実験結果を，表 4 は解析対象の動詞が IPAL の辞書と VDIC の辞書の両方に含まれている場合の実験結果を示す．

表 3 のように，われわれの手法は評価 B で 92.39%の精度を得た．村田法では評価 B で 89.09%の精度を得た．本手法は 3.30%の精度向上を実現したことを意味する．両側符合検定により有意水準 1%で本手法の方が村田法よりも優れていることを確認した．

表2　能動文への変換後の格助詞の分布

変換先の格助詞	出現率
を	33.05%
に	19.69%
と	16.00%
で	13.65%
が	11.07%
が(または)で	2.40%
から	2.13%
その他	2.01%

表3　実験結果

手法	評価 A	評価 B
ベースライン 1	58.67%	61.41%
ベースライン 2	33.05%	33.56%
KNP	27.35%	28.69%
KNP + ベースライン 1	64.32%	67.06%
KNP + ベースライン 2	48.10%	48.99%
近藤法	39.21%	40.88%
近藤法 + ベースライン 1	65.27%	68.57%
近藤法 + ベースライン 2	54.87%	56.54%
村田法	86.86%	89.09%
本手法	89.99%	92.39%

表4　解析対象の動詞が IPAL と VDIC の両方に含まれる文のみを対象とした精度

手法	評価 A	評価 B
ベースライン 1	57.71%	58.98%
ベースライン 2	37.39%	37.39%
KNP	74.59%	75.86%
近藤法	76.04%	77.50%
村田法	88.20%	89.47%
本手法	94.19%	95.46%

KNP と近藤法は性能が低い(評価 B で 28.69%と 40.88%)．これらの手法が有効に働く場合である，解析対象の動詞が IPAL の辞書と VDIC の辞書の両方に含まれている場合の表 4 の実験結果では，評価 B で 75.86%と 77.50%を得た．しかしこれらの精度は本手法に比べるとはるかに低い．またこの場合は KNP と近藤法の性能もあがるが，それらの手法の情報も利用する本手法は特に高い性能(評価 B で 95.46%)になる．

次に，素性の重要性を確かめる実験を行った．その結果を表 5に示す．表では各素性を取り除いた場合の精度を示している．実験には本稿の手法を利用した．

この結果から，F25 を使用しない場合に精度が特に下がっていることが分かる(約 1.2～1.3%の低下)．村田法と F25 を除いた時の両方で誤りであり，全素性を使用した時には正解であったものの例をあげる．

- 家族に反対される部員が多かった．(正解「が」)
  F25 なし・村田法「に」，全素性「が」
- ドゥダエフ政権登場以来，このルートを同政権に押さえられてしまった．(正解「が」)
  F25 なし・村田法「に」，全素性「が」

F25 の素性を使用しない場合や村田法では，どちらの例も格助詞が変換されず，もとの格助詞の「に」がそのまま出力されていた．一方，全素性を使用した場合では，どちらも正しい出力であった．F25 は近藤法の変換規則を利用するものだが，近藤法では格助詞「に」に関して精密な規則を作成しており，その情報も機械学習の素性として整合性よく利用できる本手法ではこれらの例を正しく解くことができた．このような素性も利用することで本手法は村田法に比べて高い精度を獲得できたものと思われる．

表 5 から，取り除くと精度の向上する素性が存在していることがわかる．このため素性選択をすることで精度が向上すると思われる．しかし素性選択の実験も試みたが素性選択の学習に利用しなかったデータの精度評価では素性選択をしてもすべての素性を利用する方法とほとんど変わらない精度であった(本稿の実験に用いたデータと異なるデータでどちらも 92.00%の精度であった)．

最後に実験データの量を変更した場合の実験を行った．その結果を図 4 に示す．実験には本稿の手法を利用し，全データの 1/2, 1/4, 1/8, 1/16 のデータで精度を求めた．データ量が半分の時に比べ，すべてのデータを用いた場合の精度が 1.78%高いことから，データ量を増やすことで精度を向上させる余地がまだ残っていると思われる．

表5　各素性を除いた場合の精度の変化

	評価A		評価B	
除いた素性	精度	差分	精度	差分
全素性使用	89.99%	---	92.39%	---
F1	89.99%	0.00%	92.39%	0.00%
F2	89.82%	-0.17%	92.28%	-0.11%
F3	89.82%	-0.17%	92.28%	-0.11%
F4	89.65%	-0.34%	92.45%	0.06%
F5	90.04%	0.05%	92.51%	0.12%
F6	90.04%	0.05%	92.45%	0.06%
F7	89.43%	-0.56%	91.95%	-0.44%
F8	89.77%	-0.22%	92.11%	-0.28%
F9	89.99%	0.00%	92.34%	-0.05%
F10	90.16%	0.17%	92.56%	0.17%
F11	90.77%	0.78%	92.95%	0.56%
F12	90.04%	0.05%	92.51%	0.12%
F13	89.65%	-0.34%	92.06%	-0.33%
F14	89.88%	-0.11%	92.28%	-0.11%
F15	89.77%	-0.22%	92.17%	-0.22%
F16	89.93%	-0.06%	92.34%	-0.05%
F17	89.88%	-0.11%	92.28%	-0.11%
F18	90.21%	0.22%	92.62%	0.23%
F19	90.04%	0.05%	92.45%	0.06%
F20	90.21%	0.22%	92.62%	0.23%
F21	89.99%	0.00%	92.39%	0.00%
F22	90.10%	0.11%	92.51%	0.12%
F23	89.93%	-0.06%	92.39%	0.00%
F24	89.99%	0.00%	92.45%	0.06%
F25	88.70%	-1.29%	91.16%	-1.23%
F26	90.04%	0.05%	92.45%	0.06%
F27	90.32%	0.33%	92.73%	0.34%
F28	89.82%	-0.17%	92.23%	-0.16%
F29	89.99%	0.00%	92.39%	0.00%
F30	89.99%	0.00%	92.39%	0.00%
F31	89.88%	-0.11%	92.28%	-0.11%
F32	89.99%	0.00%	92.39%	0.00%

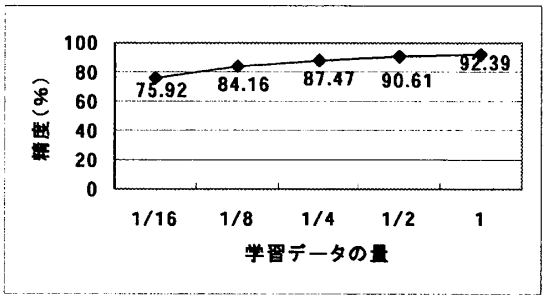

図4　学習データの量と精度の関係

5. おわりに

本研究では，日本語の受け身文を能動文に変換する際に変更されるべき格助詞を機械学習を用いて自動変換した．数量的な実験の結果，92.39%の精度を得た．先行研究の手法とも比較実験を行い，先行研究の手法の精度が80%弱であり，我々の手法の方が性能が高いことを確認した．また，我々は文献[7]において，同様に機械学習手法に基づいた格助詞の変換の研究を行っていたが，本研究ではこの研究に比べて機械学習で用いる素性を充実させることで3.3%(89.09%から92.39%)の精度向上を実現した．受け身文の能動文への変換は生成，言い換え，知識獲得，質問応答システムと数多くの分野で役に立つものである．将来，われわれはこの研究の成果を種々の自然言語処理に利用するつもりである．

参考文献

[1] Masaki Murata and Hitoshi Isahara, Universal model for paraphrasing --- using transformation based on a defined criteria ---, *NLPRS'2001 Workshop on Automatic Paraphrasing: Theories and Applications*, (2001).

[2] 乾健太郎, テキスト簡単化における聾者向け読解支援 --- 現状と展望 ---, 電子情報通信学会 言語理解とコミュニケーション研究会 WIT00-34, (2000).

[3] 村田真樹, 内山将夫, 井佐原均, 類似度に基づく推論を用いた質問応答システム, 自然言語処理研究会 2000-NL-135, (2000), pp. 181--188.

[4] 情報処理振興事業協会技術センター, 計算機用日本語基本動詞辞書 IPAL (Basic Verbs) 説明書, (1987).

[5] Sadao Kurohashi and Makoto Nagao, A Method of Case Structure Analysis for Japanese Sentences based on Examples in Case Frame Dictionary, *IEICE Transactions on Information and Systems*, Vol. E77--D, No.2, (1994), pp. 227--239.

[6] 近藤恵子, 佐藤理史, 奥村学, 格変換による単文の言い換え, 情報処理学会論文誌, Vol.42, No.3, (2001).

[7] 村田真樹, 井佐原均, 受け身／使役文の能動文への変換における機械学習を用いた格助詞の変換, 情報処理学会 自然言語処理研究会 2002-NL-149, (2002).

[8] Taku Kudoh and Yuji Matsumoto, Use of support vector learning for chunk identification, *CoNLL-2000*, (2000), pp. 142--144.

[9] 平博順, 春野雅彦, Support vector machine によるテキスト分類における属性選択, 情報処理学会論文誌, Vol.41, No.4, (2000), pp. 1113--1123.

[10] Masaki Murata, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara, Using a support-vector machine for Japanese-to-English translation of tense, aspect, and modality, *ACL Workshop on the Data-Driven Machine Translation*, (2001).

[11] Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, (Cambridge University Press 2000).

[12] Taku Kudoh, TinySVM: Support Vector Machines, http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/index.html, 2000).

[13] 工藤拓, 松本裕治, Support vector machine を用いた chunk 同定, 自然言語処理研究会 2000-NL-140, (2000).

[14] 黒橋禎夫, 日本語構文解析システム KNP 使用説明書 version 2.0b6, (京都大学大学院情報学研究科, 1998).

[15] Hans van Halteren, Jakub Zavrel and Walter Daelemans, Improving accuracy in word class tagging through the combination of machine learning systems, *Computational Linguistics*, Vol.27, No.2, (2001), pp.199--229.