

構造学習を用いた述語項構造解析

平 博順 永田 昌明

NTT コミュニケーション科学基礎研究所

taira@cslab.kecl.ntt.co.jp nagata.masaaki@lab.ntt.co.jp

1. はじめに

近年、言語処理の分野では述語項構造解析が注目を浴びている。述語項構造解析は、述語と項（日本語では述語と格関係にある単語）との関係を同定するもので、情報抽出、自動要約、機械翻訳など広範囲のテキスト処理のタスクにおいて、意味処理による精度向上を考えたときにキーとなると考えられている要素技術である。

述語項構造解析に関連する技術としては、1990年代においても、主に機械翻訳の分野で人手で作成したルールを用いて格解析を行う例があったが、それほど汎用性の高いものではなかった。近年、英語を対象とした述語項構造解析に関しては、PropBank [10]、NomBank [9] といった動詞、名詞における述語項構造の辞書が構築されるなど、活発に研究されてきている。また、日本語においても、インターネット上のテキストからの大規模な格フレームの自動構築 [3] が試みられるとともに、述語項構造解析のためのコーパス整備が進められている [2, 4]。そこでわれわれは、これらのコーパスの一つである NAIST テキストコーパスを用いて述語項構造解析を試みた。日本語の述語項構造解析を、単語間の係り受け構造から述語項構造への構造変換の問題であることから、構造学習手法の適用を試みた。構造変換を扱うことができるアルゴリズムとしては Voted Perceptron [1]、MIRA [8] など多くのアルゴリズムが提案されている。われわれはそれらの中で SVM^{struct} [11] を採用した。 SVM^{struct} は、マージン最大化学習アルゴリズムの一つで最適性において優れているという特徴がある。実際、固有表現認識タスクなどで、Voted Perceptron や条件付確率場 (CRF) [7] 等の手法より高精度の結果が得られており [11]、述語項構造解析においても高い認識精度が得られる可能性が高いと考え適用を試みた。また、そこで得られた変換ルールを用いて未知のテキストに対する解析精度の評価を行った。

2. 構造学習による述語項構造解析

本節では、2.1 節で SVM^{struct} の概要を説明した後、2.2 節で述語項構造解析への適用方法について述べる。

2.1. SVM^{struct} の概要

入力構造 $x \in \mathcal{X}$ から出力構造 $y \in \mathcal{Y}$ への写像を考え、訓練用正解データとして、以下のように m 個の入出力ペア

$$(x^1, y^1), \dots, (x^m, y^m) \in \mathcal{X} \times \mathcal{Y}$$

が与えられたとする。ここで、 \mathcal{X} および \mathcal{Y} は取りうる可能性のある構造の集合であり、フラット素性集合だけでなく、ツリー構造、グラフ構造といった複雑な構造も原理的には扱うことができる。

SVM^{struct} では、入力構造と出力構造の組み合わせに対し、正しい変換の組み合わせであるときに大きな値をとる関数 $h: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ を考え、与えられた訓練用正解データを用いて最適な h を求めることを学習の目的とする。また、この関数を得られれば、未知のデータ x^{test} に対する $y^{predict}$ は以下のように求められる。

$$y^{predict} = \arg \max_{y \in \mathcal{Y}} h(x^{test}, y; \mathbf{w})$$

ここで \mathbf{w} は関数 h に含まれる、 x と y の組み合わせに対する重みパラメータである。関数 h の定義の仕方は様々なものが考えられるが文献 [11] では簡単のため x と y の組に対して一つの実数値を返す関数 $\Psi(x, y)$ を設定し、この関数と重み \mathbf{w} との内積としている。すなわち

$$h(x, y; \mathbf{w}) = \langle \mathbf{w}, \Psi(x, y) \rangle$$

と定義している。

次にパラメータ \mathbf{w} を決定するための学習を、マージン最大化の考えに基づいて行う。 SVM^{struct} では正解入出力ペア (x^i, y^i) と不正解入出力ペア $(x^i, \bar{y}^{i,k})$ との間のマージンを最大にすることを目的とする。ここで、

$\bar{y}^{i,k}$ は入力 x^i に対する不正解出力のうち k 番目のものを表す。すなわち、

$$\begin{aligned} & \max (h(x^i, y^i; \mathbf{w}) - h(x^i, \bar{y}^{i,k}; \mathbf{w})) \\ & = \max (\mathbf{w} \cdot \Psi(x^i, y^i) - \mathbf{w} \cdot \Psi(x^i, \bar{y}^{i,k})) \\ & = \max \mathbf{w} \cdot \delta \Psi^{i,k} \end{aligned}$$

を満たす \mathbf{w} を求める。ここで $\delta \Psi^{i,k}$ は、

$$\delta \Psi^{i,k} \equiv \Psi(x^i, y^i) - \Psi(x^i, \bar{y}^{i,k})$$

と定義した。

SVM のときと同様にソフトマージンの考え方を取り、訓練正解データに対しエラーを許し、

$$\begin{aligned} \langle \mathbf{w}, \delta \Psi^{i,k} \rangle & \geq 1 - \xi_i, \quad \xi_i \geq 0, \\ i & = 1, \dots, m, \quad \forall k \end{aligned}$$

という制約にする。ここで、 ξ_i は i 番目の訓練正解データに対応したスラック変数である。また、マージンの最大化と m 個の訓練データエラーの考慮の配分を $\frac{C}{m}$ とし、

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{i=1}^m \xi_i$$

の最適化を行う。ここで考慮の配分が SVM のように C でなく、 $\frac{C}{m}$ であるのは、そうすることで後の式が簡単になるためである。この最適化問題に対する双対問題は、

$$\begin{aligned} L(\mathbf{w}, \xi, \alpha) & \\ & = \sum_{i=1}^m \sum_{k=1}^K \alpha_{i,k} - \frac{1}{2} \sum_{i=1}^m \sum_{k=1}^K \sum_{i'=1}^m \sum_{k'=1}^K \alpha_{i,k} \alpha_{i',k'} \mathbf{x}^{i,k} \mathbf{x}^{i',k'} \\ & \quad 0 \leq \alpha_{i,k} \\ & \quad m \sum_{k=1}^K \frac{\alpha_{i,k}}{\Delta(y^i, \bar{y}^{i,k})} \leq C \end{aligned}$$

となる。ここで、 L はラグランジュ関数、 $\alpha_{i,k}$ は i 番目の訓練データにおける不正解ペア $(x^i, \bar{y}^{i,k})$ の制約に対応するラグランジュ乗数である。また、 $\Delta(y^i, \bar{y}^{i,k})$ は、構造 y^i と構造 $\bar{y}^{i,k}$ の間での構造の異なり度合いを表す関数で、タスクに応じて適切な関数をユーザが設定する。ただし、同じ構造同士に対する値は 0 を取るものとする。

SVM^{struct} は、この最適化問題を解くためのヒューリスティクスを含めた枠組みの総称である。 SVM^{struct} のページ¹ においてソフトウェアが提供されているが、実際には、タスクに依存してかなり内部まで実装が必要となるため、われわれは上記のアルゴリズムを参考にして最適化部分も含め、独自に実装を行った。また、学習で得られたモデルを用いて未知データを評価するデコーダについても実装を行った。

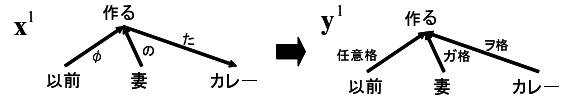


図 1. 格変換の例

$$\mathbf{x}^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{array}{l} \leftarrow \text{"fc, 以前, } \phi \\ \leftarrow \text{"fc, 妻, の"} \\ \leftarrow \text{"bc, カレー, た"} \end{array} \quad \mathbf{y}^1 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \begin{array}{l} \text{任意格: 0} \\ \text{ガ格: 1} \\ \text{ヲ格: 2} \\ \text{二格: 3} \end{array}$$

図 2. ベクトル化の例

2.2. 述語項構造解析への適用

次に上記の学習を述語項構造解析に適用する方法について述べる。例として「以前、妻の作ったカレーをおいしく食べていたとき、…」という文の先頭部分「以前、妻の作ったカレー」について考える。述語「作った」の基本形に対し、「妻」が「ガ格」、「カレー」が「ヲ格」を与えるという正解が与えられているものとする。また、この例文に対し、係り受け解析を行った結果、文節「以前、」および文節「妻の」が文節「作った」に、文節「作った」が文節「カレー」に係っていることが解析できたとする。この時、係り受け構造 x^1 から述語項構造 y^1 への変換を図 1 のようにとらえる。すなわち、入力構造に関しては、述語「作る」に対して係り受け関係にある文節の意味主辞を項の候補とし、述語と意味主辞との間のラベル付き有向グラフととらえる。ここでのラベルは意味主辞に後続する付属語とする。また意味主辞は、係り受け解析器（本稿の実験では Cabocha²）がその文節の主辞として判定した単語と定義する。また、出力構造に関しては、候補となる項と述語との間のラベル付き有向グラフととらえる。ここでラベルは正解の格の種類とする。

次に、計算機で扱いやすくするために、入力構造 x^1 および出力構造 y^1 をそれぞれベクトル化する。その様子を図 2 に示す。ここで、入力構造 x に関しては、（意味主辞と述語の間での係り受けタイプ、意味主辞、付属語）の三つ組を一つの素性とし、その組み合わせが出現した場合を 1、それ以外の場合を 0 の値を与えるとする。係り受けタイプに関しては、意味主辞から述語へ係るタイプ（「fc」で表す）、述語から意味主辞へ係るタイプ（同「bc」）、所属文節が同一文節のタイプ（同「sc」）、同一文内で係らない、もしくは全く係り受け関係にない

¹http://svmlight.joachims.org/svm_struct.html

²<http://chasen.org/taku/software/cabocha/>

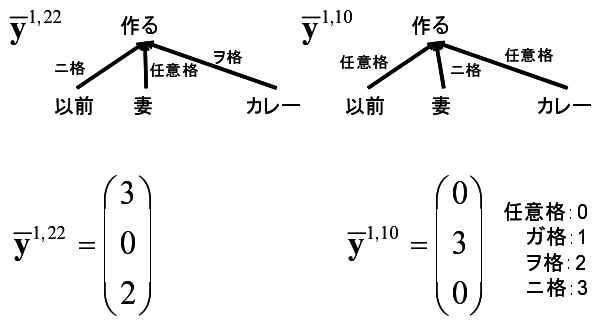


図 3. 不正解構造の生成

タイプ (同「nc」) の 4 種類とした。また、付属語がない場合は「 」として表現した。他方、出力構造 y に関しては、変換前の構造に対して、何格に変換されたかによって 0~3 の値を与える。この図では、本稿で使った NAIST テキストコーパスに準じて、ガ格、ヲ格、ニ格の 3 つの必須格とそれ以外の任意格の格の違いによって値を与えているが、実際には、それ以外の格についても同様に設定することができる。これらのベクトル化された正解構造を構造学習器 SVM^{struct} への入力とする。

構造学習器 SVM^{struct} の中では、ベクトル化された変換後の正解構造に対して、不正解の構造を生成して学習に用いる必要がある。入力された入力構造 x に対して正解以外に取りうる構造は、タスクによって異なるため、不正解構造の生成についてはタスク毎に実装する必要がある。変換後の構造として、N-best の解が得られているようなタスクではそれらを用いればよいが、本稿で扱う述語項構造解析では、現存する述語項構造コーパスがそうであるように 1 位正解が与えられない状況を考えるため、不正解構造の自動的な生成を行った。必須格に関しては、複数の単語が同時に同じ格スロットに入らないという制約があるため、その制約を考慮した不正解構造を生成する。例えば図 1 の正解構造 y^1 に関しては、合計 32 種類の不正解構造が考えられる。図 3 にそのうち 2 種類の例を示した。

先に述べたように、これら得られた不正解構造に対して、正解構造との構造の異なり度合いを表す $\Delta(y^i, \bar{y}^{i,k})$ も求めておく。本稿では、 $\Delta(y^i, \bar{y}^{i,k}) = (\text{格変換が間違った個数})$ とした。

以上に述べたような方法で、日本語述語項構造解析に対する SVM^{struct} を実装し、変換モデルの学習を行った。また、得られたモデルのもとで目的関数が最大となるような構造 y^1 を求めるデコーダについても独自に実装を行った。

3. 実験

3.1. 実験データ

本計算機実験では、述語項構造解析を構造学習を用いて行うための訓練用データ、評価用データを、NAIST テキストコーパス Ver 1.4³ で付与されている述語項構造正解から生成した。NAIST テキストコーパスは、京都テキストコーパス Ver 3.0 [6] 全記事の 2,929 記事、38,384 文に対し、106,628 個の述語および 28,569 個の事態性名詞に対し、特に必須表層格 (ガ、ヲ、ニ) についての述語項構造および共参照関係にタグを付与されたコーパスである [2]。

このコーパス全体について、コーパスに現れる述語および事態性名詞の中からランダムに 200 単語を選び、それらの述語および事態性名詞それぞれに対し評価用データを作成した。多くの構造学習器同様、 SVM^{struct} についても計算に大きなコストがかかるため、何らかの計算コスト削減の工夫が必要となるが、現時点ではそこまでの実装が十分でないため、サンプル数を削減して実験を行った。社説記事 1 月-12 月、記事 1 月 1 日、3 日-17 日の順に述語項構造の正解が付与されている文を抽出し、述語・事態性名詞毎に最初の 5 サンプルを訓練データ、その後の 5 サンプルをテストデータとして、合計、訓練データ 500 サンプル、テストデータ 500 サンプルのデータを作成した。また、文中に出現した任意格については、長い文になると多数の任意格が現れ計算量が增大するため、文中で最初に出現する 5 単語までを学習で考慮した。また、訓練データ数を減らしたことから、項となる単語の網羅性が低くなるため、本実験では、使用する意味主辞については、その品詞についての構造変換として、実験を行った。ただし、有向グラフのラベルに相当する付属語に関しては、品詞ではなくその基本形をそのまま使用した。また、文節同士の係り受け情報は日本語係り受け解析器 Cabocha [5] の出力結果を正解として使用した。

3.2. ベースライン手法

本実験では、提案手法に対する比較するベースラインとなる手法として、項の述語・事態性名詞に対する係り受け関係、それぞれの組み合わせに対し、訓練データに現れた正解の格で最も頻度の高い格を出力とするモデルを作成し、テストデータを評価した。

3.3. 実験結果と考察

上記、訓練データ 500 サンプルに対し、 SVM^{struct} で学習、テストデータ 500 サンプルで評価した結果の正解率 (%) を表 1 に示す。評価については、格の同定が

³<http://cl.naist.jp/nldata/corpus/>

比較的難しいヲ格、二格に対しておこなった。ヲ格、二格とも訓練データ、テストデータ双方について、精度向上が見られた。

表 1. 評価実験の結果

	訓練データ		テストデータ	
	ヲ格	二格	ヲ格	二格
ベースライン手法	96.36	98.51	45.12	46.62
提案手法	99.51	99.48	46.15	47.97

4. おわりに

日本語における述語項構造解析を単語間の係り受け構造から、述語項構造への構造変換ととらえ、構造学習手法の一つである SVM^{struct} を用いて変換ルールの学習を試みた。機械学習器およびデコーダは独自に実装を行い、評価用データとして NAIST テキストコーパスから作成した小規模な評価データを用いて評価実験を行った。網羅性の低い小規模な学習データながら、構造学習を使わないベースライン手法に比べ、若干の精度向上が見られた。今後、学習器を改良し、より大規模なデータに対し学習を行い、手法の有効性を確認する予定である。また、今回用いた品詞に加え日本語語彙大系 [12] などのシソーラスを用いた意味属性に関するモデル化も試みたいと考えている。

謝辞

NAIST テキストコーパスの仕様に関して貴重なコメントを頂きました、奈良先端科学技術大学院大学の飯田龍氏に感謝いたします。

参考文献

- [1] Collins, M. and Duffy, N.: New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and Voted Perceptron, *Proc. of 40th Annual Meeting of the Association for Computational Linguistics(ACL2002)* (2002).
- [2] 飯田龍, 小町守, 乾健太郎, 松本裕治: NAIST テキストコーパス: 述語項構造と共参照関係のアノテーション, 情報処理学会研究報告 (自然言語処理研究会) NL-177-10, pp. 71-78 (2007).
- [3] 河原大輔, 黒橋禎夫: 高性能計算環境を用いた Web からの大規模格フレーム構築, 情報処理学会研究報告 (自然言語処理研究会) NL-171-12, pp. 67-73 (2006).
- [4] 河原大輔, 黒橋禎夫, 橋田浩一: 「関係」タグ付きコーパスの作成, 言語処理学会第 8 回年次大会発表論文集, pp. 495-498 (2002).

- [5] 工藤拓, 松本裕治: チャンキングの段階適用による日本語係り受け解析, 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834-1842 (2006).
- [6] 黒橋禎夫: 京都テキストコーパス Version 4.0.
- [7] Lafferty, J., McCallum, A. and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proc. of the 18th International Conference on Machine Learning(ICML-2001)*, pp. 282-289 (2001).
- [8] McDonald, R., Crammer, K. and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics(ACL2005)*, pp. 91-98 (2005).
- [9] Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B. and Grishman, R.: The NomBank Project: An Interim Report, *Proc. of HLT-NAACL 2004 Workshop on Frontiers in Corpus Annotation* (2004).
- [10] Palmer, M., Kingsbury, P. and Gildea, D.: The Proposition Bank: An Annotated Corpus of Semantic Roles, *Computational Linguistics*, Vol. 31, No. 1, pp. 71-106 (2005).
- [11] Tsochantaridis, I., Hofmann, T., Joachims, T. and Al-tun, Y.: Support Vector Machine Learning for Interdependent and Structured Output Spaces, *Proc. of ICML2004*, pp. 823-830 (2004).
- [12] 池原 悟, 宮崎正弘, 白井 諭, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林 良彦: 日本語語彙大系, 岩波書店 (1997).