

B4IM2030

修士論文

ニューラルネットに基づく日本語係り受け解析

田中亮太

2016年2月17日

東北大学 大学院  
情報科学研究科 システム情報科学専攻

本論文は東北大学 大学院情報科学研究科 システム情報科学専攻に  
修士(工学) 授与の要件として提出した修士論文である。

田中亮太

審査委員：

乾 健太郎 教授 (主指導教員)

木下 賢吾 教授

伊藤 彰則 教授

岡崎 直観 准教授 (副指導教員)

# ニューラルネットに基づく日本語係り受け解析\*

田中亮太

## 内容梗概

日本語の係り受け解析では、二文節間の係り受け関係を解くために、文節間の基本的な素性に加え、それらの複雑な組み合わせを獲得する必要がある。これまで組み合わせ素性は、有効だと思われる組み合わせの人手による列挙や、非線形カーネルを用いた手法により獲得されてきた。しかしながら人手による列挙には高度な言語学的知識が必要とされ、また非線形カーネルを用いた手法は学習・解析に非常に時間が掛かるという欠点がある。

入力間の複雑な組み合わせ素性を獲得できる手法として、ニューラルネットが挙げられる。近年、英語における NLP 分野では、複雑な構造を持つニューラルネットにより、従来のモデルでは捉えきれない素性構造を獲得し、精度を上げるとする多くの研究がなされている。しかしながら、これまで日本語自然言語処理に対してニューラルネットを適用した研究は殆どなされてこなかった。そこで本研究では、日本語係り受け解析にニューラルネットを適用する事を検討する。

本研究ではまず、人手により設計された組み合わせ素性に対して、ニューラルネットがより優れた組み合わせを獲得できるかについての検証を行った。実験結果から、適切な学習パラメータ及び学習法を用いることで、ニューラルネットが人手による手法より優れた精度を達成できることを示した。また、学習を終えたニューラルネットワークの中間層を解析することにより、組み合わせ素性が正しく獲得されているかの調査を行った。

また、本研究が用いたモデルに近い構造を持つ、英語における先行研究にならない、単語分散表現を導入した場合の効果について、具体的な検証を行った。

## キーワード

自然言語処理, 係り受け解析, 組み合わせ素性, ニューラルネット

---

\*東北大学 大学院情報科学研究科 システム情報科学専攻 修士論文, B4IM2030, 2016年2月17日.

# Contents

<b>1</b>	<b>はじめに</b>	<b>1</b>
1.1	研究背景 . . . . .	1
1.2	本研究の目的 . . . . .	2
1.3	本研究の構成 . . . . .	2
<b>2</b>	<b>関連研究</b>	<b>3</b>
2.1	日本語係り受け解析 . . . . .	3
2.2	基本素性と組み合わせ素性 . . . . .	4
2.3	ニューラルネットと係り受け解析 . . . . .	4
<b>3</b>	<b>ニューラルネットによる係り受け解析</b>	<b>5</b>
3.1	係り受け解析と組み合わせ素性 . . . . .	5
3.2	フィードフォワード型ニューラルネットワーク . . . . .	6
3.3	モデル概要 . . . . .	7
3.4	係り受け解析モデル . . . . .	9
<b>4</b>	<b>既存研究との比較と分析</b>	<b>10</b>
4.1	内元ら [1] との比較 . . . . .	10
4.1.1	ベースライン . . . . .	10
4.2	使用する基本素性 . . . . .	12
4.2.1	実験設定 . . . . .	13
4.2.2	実験結果 . . . . .	13
4.3	学習された重みパラメータの分析 . . . . .	15
4.4	工藤ら [2] との比較 . . . . .	17
4.4.1	ベースライン . . . . .	17
4.4.2	素性設定及び実験設定 . . . . .	17
4.4.3	実験結果 . . . . .	18
<b>5</b>	<b>単語分散表現の導入の検討</b>	<b>20</b>
5.1	単語分散表現とは . . . . .	20
5.2	英語における類似モデル . . . . .	21
5.3	ニューラルネットに基づく日本語係り受け解析への単語分散表現 の導入 . . . . .	22

5.3.1	実験設定 . . . . .	22
5.3.2	実験結果 . . . . .	23
5.4	次元数を変化させた場合 . . . . .	24
<b>6</b>	<b>おわりに</b>	<b>25</b>
	謝辞	26
	付録	30
A	内本 [1] における素性設定	30

## List of Figures

1	1入力1出力のニューラルネット . . . . .	7
2	ニューラルネットの構成図 . . . . .	8
3	交差のある係り受け（上）及びない係り受け（下） . . . . .	9
4	ロジスティック回帰と3層ニューラルネット . . . . .	11
5	入力-中間層行列の発火パターン（縦軸：基本素性テンプレート，横 軸：対応する中間層） . . . . .	15
6	入力-中間層行列の発火パターン . . . . .	16

## List of Tables

1	使用した基本素性 . . . . .	12
2	学習及びテストに用いたコーパス . . . . .	13
3	内本ら [1] の素性設定における係り受け正解率及び文正解率 . . . . .	14
4	使用した基本素性 . . . . .	18
5	工藤ら [2] の素性設定における係り受け正解率及び文正解率 . . . . .	19
6	テストデータにおける各表層素性の未知語数 (未知語数/素性数)	22
7	単語分散表現の初期化に対する精度の変化 . . . . .	23
8	単語分散表現の次元数を変化させた場合 . . . . .	24
9	基本素性一覧 . . . . .	30
10	組み合わせテンプレート一覧 . . . . .	32

# 1 はじめに

本節では、まず研究背景及び目的を述べた後、本論文の構成を示す。

## 1.1 研究背景

日本語係り受け解析では、文中のある二文節の係るか・係らないかの判別を繰り返すことにより、文全体の係り受け関係を決定する。二文節の係り受けの判別を行うために、まず係り元と係り先にまつわる基本素性（係り元がどのように言い切られているか、あるいは後ろ文節の主辞となる単語は何かなど）を抽出する。しかしながら、これら基本素性だけでは係り関係の判別を行うには不十分であり、第3章で述べるように、基本素性間の複雑な組み合わせ素性を考慮する必要がある。

これまでは、組み合わせ素性の獲得を、言語学的直感に基づいた人手による記述 [1] や、多項式カーネルを用いた SVM[2][3][4][5][6]、あるいはカーネルの展開の考慮 [7][10][11] などにより対処してきた。組み合わせ素性の人手による記述は、重要だと思われる組み合わせを選択的に捉えるため、抽出される素性数を少なくできるという利点がある。しかしながら、組み合わせ素性の列挙には高い専門性を必要とし、また重要な組み合わせを人手では列挙しきれない可能性がある。非線形カーネルに多項式カーネルを用いた手法は、組み合わせ素性の獲得を非線形カーネルにより獲得しており、高い精度を達成している。しかしながら、教師データサイズ及び入力素性が大きい場合、学習に非常に時間が掛かるという欠点がある。これに対し、多項式カーネルの展開を考え、展開された膨大な組み合わせ素性の内、有効であるものを選択し、効率的に学習及び解析を行う研究も提案されている [7][9][10][11]。これらの手法では基本的に、展開された素性の内の有効なもののみを使って線形回帰を行うため非常に高速であるが、多項式カーネルの次数  $d$  を大とした場合に展開される素性数が膨大になるため、次数  $d$  を限定（通常は  $d \leq 4$ ）する必要がある。

入力間の複雑な素性の組み合わせを獲得できる他の手法として、ニューラルネットワーク（以後ニューラルネット）が挙げられる。ニューラルネットは、英語における NLP の基礎・応用タスクで近年盛んに研究され、各タスクにおいて高い性能を示すことが多く報告されている。しかしながら、ニューラルネットを日本語タスクに適用し、組み合わせ素性を獲得したとする研究はこれまでなされてこなかった。



また、NLP 分野では、対象タスクの精度を向上させるための取り組みとして、外部の大規模なコーパスで教師無し学習を行った単語分散表現を入力として加えるなどの工夫がなされてきた。近年、ニューラルネットを用いることで、単語分散表現もパラメータとして学習することで精度の向上を行う研究が報告されている [12] が、SVM などの既存手法ではこれらの単語分散表現を入力として加えることはできても、パラメータとして学習することは困難である。

## 1.2 本研究の目的

本研究ではまず、日本語係り受けタスクを解くために必要とされている組み合わせ素性を、ニューラルネットが捉えることができるかについての検証を行う。そのためにまず、人手で設計された内本 [1] らと同じ素性設定を用い、組み合わせ素性を用いたベースラインと、基本素性のみを用いたニューラルネットとの比較実験を行い、ニューラルネットが人手では捉えられない組み合わせを獲得できているかについて確かめた。この時、学習されたパラメータの内部を分析することで、どのように組み合わせ素性が獲得されるかを明らかにした。また、本モデルと同じ解析法を用いている、非線形カーネルを用いた手法に対し、同じ素性設定のもとで比較実験を行い、非線形カーネルを用いた手法と同等の性能を示すことができるかについて調査した。さらに、英語における係り受けタスクにおいて、本研究に近い構成のニューラルネットを用いた研究に対し、それらの研究が用いている単語分散表現が、実際にどのような影響を与えているのかについても実験を行い、分散表現の有用性についての検証を行った。

## 1.3 本研究の構成

本論文では、まず第2章で日本語係り受け解析及び英語におけるニューラルネットの関連研究について述べる。次に第3章で組み合わせ素性の詳細な役割とニューラルネットによる組み合わせ素性の獲得の検証を行う。さらに、第4章で既存手法との比較を行い、第5章で単語分散表現の導入を検討する。最後に、第6章で総括を行う。

## 2 関連研究

### 2.1 日本語係り受け解析

日本語係り受け解析では、まずタスクの前処理として文の形態素解析と文節まとめ上げを行い、文節列を生成する。その後、得られた文節列中のすべての文節について、係り先の文節を推定するという問題を解くことになるが、どの文節を先に解いていくかによっていくつかのバリエーションが存在する。例えば、文末から文節を見ていき、非交差条件を考慮しながら係り先を決定していく手法 [1] や、文頭から文節を見ていき、直後に係るか・係らないかによってまとめ上げていく Shift-Reduce 法 [3]、あるいはすべての文節について、直後に係るかどうかを判別し、係ると判定したものから決定していく、チャンキング段階適用法 [4] などが存在する。これらの手法では基本的に、係り元の文節と係り先の文節についての素性を抽出し、これら二分節が係るか/係らないかをモデル化し、解析を行っている。

これに対し、すべての係り先候補に対し、それらの間でどの候補が尤もらしいかということモデル化し、解析を行う手法も提案されている。相対モデル [5] では、係り元に対するすべての係り先候補に対し、係り先となる確率を相対的にモデル化している。トーナメントモデル [6] では、係り先候補をトーナメント形式で競り合わせ、どの候補が係り先として最も優れているかを選別していく。これらのモデルは、係り元・係り先の二文節間の係りやすさをモデル化する手法に対し、係り先候補間の優劣を表すように学習するため、二文節間だけではなく周辺の情報を組み入れることができ、良い性能を示すことが報告されている。しかしながら、相対モデルでは係り先候補が可変であるためにモデルの拡張が難しく、トーナメントモデルはトーナメントを進める順序（前から行うか後から行うか）によって性能が変化するため、もっともよいモデルを得るためにすべての解析順序を試す必要があるという欠点がある。

その他、すべての文節とそれらの可能な係り関係をそれぞれノード・エッジとしてグラフ化し、エッジに対して係りやすきのスコアを算出後、最大部分木を見つめる MST [8] などが存在し、日本語に限らず係り受け解析タスクで広く利用されている。

本研究では、人手により組み合わせ素性した内本らの手法 [1] と比較するため、文末から貪欲に係り先を同定していく解析法を用いる。

## 2.2 基本素性と組み合わせ素性

第3章で説明するように、係り受け解析には二文節間から抽出した基本素性だけでなく、それらの複雑な組み合わせを考慮した学習・解析を行うことが必要となる。チャンキング段階適用法や Shift-Reduce 法などの、二文節に係るか・係らないかの二値分類を行うモデルでは、非線形カーネルの一種である多項式カーネルを用いた SVM により、基本素性のみから組み合わせ素性の考慮が可能である。しかしながら、一般的に非線形カーネルを用いた SVM は、トレーニングセットが多い場合に学習及び解析に膨大な時間を要するという欠点がある。

これに対し、多項式カーネルについて、その明示的な展開を考慮することで処理時間の問題を解決しようとする手法も存在する [9][10][11]。これらの手法では、多項式カーネルの明示的な展開を考え、展開される膨大な組み合わせの中から出現頻度の低いものを除き、効率的な学習と重みの計算により、処理時間の大幅な短縮に成功している。しかしながら、多項式カーネルの展開により生成される組み合わせ素性の数は膨大となるため、これらの研究ではカーネルの次数を低い次数 ( $d = 3$ ) に限定している。

## 2.3 ニューラルネットと係り受け解析

入力間の複雑な組み合わせを考慮することのできる他の解析手法として、ニューラルネットが挙げられる。ニューラルネットを自然言語処理に応用する研究はこれまで多くなされてきたが、日本語の係り受け解析に応用するとする研究は未だなされていない。

また、日本語の係り受け解析と対比されるタスクである、英語の依存構文解析 (Dependency Parsing) では、ニューラルネットを用いたシステムにより、既存手法に並ぶ精度を得られたとする研究が、最近となって多数報告されている [13][14][15]。しかしながら、これらの研究ではニューラルネットと同時に、第5章で詳細に解説するエンベディングと呼ばれる手法が使われており、システムの貢献がニューラルネットによるものか、エンベディングによるものかが明らかではなかった。

本研究では、ニューラルネットを用いた日本語係り受け解析を行うにあたって、これらのエンベディングの導入を行って効果を検証し、その有用性を確かめた。

### 3 ニューラルネットによる係り受け解析

本章では、まず組み合わせ素性の重要性及び用いた基本素性についての解説を行い、その後解析に用いるニューラルネットモデルの具体的な説明を行う。

#### 3.1 係り受け解析と組み合わせ素性

これまで何度か触れてきたように、係り受け解析は、基本素性だけでは解くことができず、それらの複雑な組み合わせ素性を考慮することが必要となる。例えば次の文について考える。

太郎は 望遠鏡で、 遠くの 風景を 眺めた

この例では、「遠くの」という文節は「風景を」という文節に、それ以外の文節はすべて「眺めた」に係る。もし、基本素性のみでこの文節を決定しようとした場合、問題が生じる。日本語の文節は、統計的に直後の文節に係りやすいため、すべての文節が「文節間距離が1」という素性のみを頼りに直後に係ってしまう。

これに対応するためには、二つ以上の基本素性の組み合わせ（＝組み合わせ素性）の考慮が必要となる。例えば文節「太郎は」は、語形に格助詞「は」を持つため、主辞が動詞のものにかかりやすく、この候補中では「眺めた」に最も係りやすい。この場合、係り元語形が格助詞であり、係り先主辞が動詞であるという2つの素性の組み合わせにより判別が可能となる。また「望遠鏡で、」などのように文節中に読点が存在する場合には、直後の文節には係りにくいという傾向があり、これは係り元読点の有無と文節間距離という2つの素性の組により判別できる。このように、係り受け解析を解くためには、2つ以上の素性の組み合わせを考慮しなければ、精度を上げることはできない。

これまでの研究では、組み合わせ素性の獲得は、非線形カーネルを用いたSVMに頼っていた[3][6][7]。その他の研究として、内元ら[1]は係り受け解析に有効と思われる2～5つまでの組み合わせ素性を人手で選定し、最大エントロピー法により学習を行っている。また、吉永ら[10]は、多項式カーネルの展開により明示的な素性の組み合わせを考慮し、よく出現するものについてトライ木を使って効率的に重みを保持している。

### 3.2 フィードフォワード型ニューラルネットワーク

ニューラルネットワークは、その構造の違いにより幾つかの種類が存在するが、最も単純な構成は、入力層・中間層・出力層を持つ3層のフィードフォワード型ニューラルネットワークである。出力層が1ノードの場合の3層のフィードフォワード型ニューラルネットワークは、システム全体を多変数入力・一変数出力の関数と見た時、中間層の素子数を増やすことで任意の関数近似能力を得ることができることで知られている [16].

例として1入力1出力、中間層  $N$  次元の場合の3層ニューラルネットを図1に示す。システムへの入力  $x$ 、及びパラメータである入力重みベクトル  $\mathbf{w} = (w_1, w_2, \dots, w_N)$ 、バイアス  $\mathbf{b} = (b_1, b_2, \dots, b_N)$ 、出力層重みベクトル  $\mathbf{w}' = (w'_1, w'_2, \dots, w'_N)$  とすると、出力  $y$  は次のように書き表せる。

$$y = \sum_{i=1}^N w'_i \cdot f(w_i x + b_i) \quad (1)$$

ここで、 $f$  は活性化関数と呼ばれ、中間ノードの出力値を変換する関数である。活性化関数には通常、シグモイド関数やハイパボリックタンジェントなどの非線形の関数を用いられるが、入力値が連続値を取る場合はRadial Based Function (RBF) が用いられることもある [17]。活性化関数に非線形関数を用いることで、入力と出力の非線形な関係を捉える事が可能となる。入力が  $M$  次元  $\mathbf{x} \in R^M$  の場合、(1) 式は、

$$y = \sum_{i=1}^N w'_i \cdot f(\mathbf{w}_i^T \mathbf{x} + b_i) \quad (2)$$

と表せる。ここで、 $\mathbf{w}_i$  は中間ノード  $i$  における入力重みベクトルであり、行列として表せば、 $W \in R^{M \times N}$  となり、

$$y = \mathbf{w}'^T f(W\mathbf{x} + \mathbf{b}) \quad (3)$$

を得る。パラメータである入力重み行列  $W$ 、出力ベクトル  $\mathbf{w}'$ 、及びバイアスベクトル  $\mathbf{b}$  は、誤差逆伝搬法を用いて勾配を計算することができる。

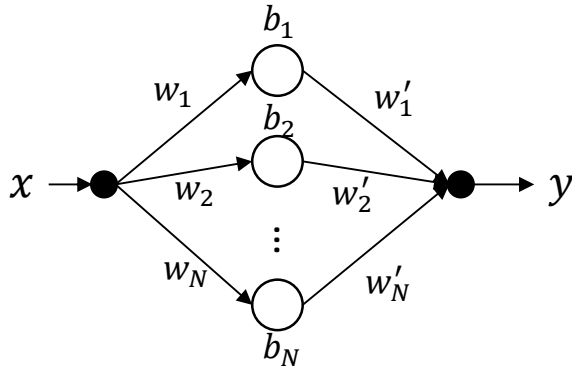


図 1: 1 入力 1 出力のニューラルネットワーク

### 3.3 モデル概要

本研究では，ニューラルネットワークの出力層を 1 次元として，二文節間の係る確率  $p$  の直接モデル化を行う．ニューラルネットワークの入力層を準備するための前処理として，基本素性の one-hot ベクトル化を行う．one-hot ベクトルとは，ベクトル中の一つの次元のみ 1 で，それ以外は 0 となるベクトルである．抽出した素性に対し，得られた素性値に対応する箇所だけ 1，それ以外を 0 として one-hot ベクトルを作成する．one-hot ベクトルの各次元は，学習コーパス中で  $n$  回以上出現した素性値の数 + 1 だけ用意し，該当する素性が無かった場合に 1 となる次元を用意する．

例えば，係り元主辞表層形  $f_s$  が，学習コーパス中で  $n$  回以上 { 日本, 示す } という素性値で出現したとする．この時， $f_s$  の one-hot ベクトルは

$$\mathbf{v} = (f_{s:\text{日本}}, f_{s:\text{示す}}, f_{s:\text{NONE}}) \quad (4)$$

の様に表される．ただし， $f_{s:\text{日本}} \in \{0, 1\}$  は， $(f_s = \text{日本})$  の時に 1，そうでない場合 0 である． $f_{s:\text{NONE}}$  は，解析時に  $f_s$  が { 日本, 示す } の何れでもない場合に 1 となる．その他の素性テンプレートについても同様に one-hot ベクトル化を行う．その後，得られた one-hot ベクトルをすべて結合し，ニューラルネットワークの入力ベクトル  $\mathbf{v} \in R^N$  を作成する．

作成した  $N$  次元入力ベクトル  $\mathbf{v} \in \{0, 1\}^N$  から， $D$  次元の中間層ベクトル  $\mathbf{h} \in R^D$  を作成する．

$$\mathbf{h} = \tanh(W_1 \mathbf{v} + \mathbf{b}_1) \quad (5)$$

ここで  $W_1$  は入力重み行列,  $\mathbf{b}_1$  は入力バイアスペクトルである. また, 活性化関数にはハイポリックタンジェントを用いる.

$W_1$  は, 例えば入力・中間層がそれぞれ30000, 200次元であれば,  $W_1 \in R^{200 \times 30000}$ ,  $\mathbf{b}_1 \in R^{200}$  となり, 30000次元の入力ベクトルを200次元に落とす線形変換を行う. 組み上げた中間層ベクトル  $\mathbf{h}$  と, 重みベクトル  $\mathbf{w}_2$  との内積を取り, 出力値  $y \in R$  を得る.

$$y = \mathbf{w}_2^T \mathbf{h} \quad (6)$$

さらに, 得られた出力値  $y$  に, シグモイド関数を通して0~1の確率値  $p$  を得る.

$$\begin{aligned} p &= \text{sigmoid}(y) \\ &= \frac{1}{1 + \exp(-y)} \end{aligned} \quad (7)$$

このスカラ値  $p$  を二文節に係る確率とする. 概要を図2に示す.

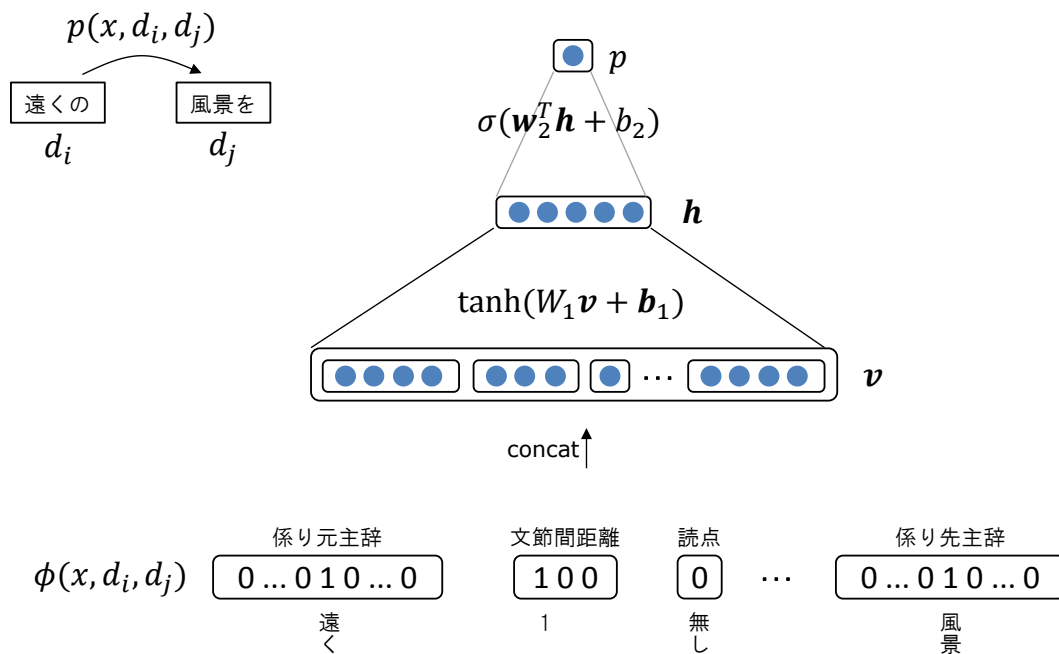


図 2: ニューラルネットの構成図

損失関数には，クロスエントロピー関数を用いる． $i$  番目の学習サンプルにおける教師信号を  $t_i \in \{0, 1\}$ ，入力ベクトルを  $\mathbf{v}_i$  と表すと，損失関数は次のように表せる．

$$L = - \sum_{i=1}^K t_i \log(p_i) + \lambda \sum_{\theta \in \Theta} \theta^2 \quad (8)$$

ここで， $p_i$  は式 (7) で得られた，入力ベクトル  $\mathbf{v}_i$  に対するモデルの出力した確率値である．また，右辺二項目は L2 正則化項であり， $\lambda$  は正則化重み係数， $\Theta$  はモデルの全パラメータである．この損失  $L$  をパラメータ  $\Theta$  について偏微分を行い，得られた勾配を用いて  $L$  の最小化を行う．学習勾配の計算には AdaGrad[18] を用い，ミニバッチによる学習を行う．

### 3.4 係り受け解析モデル

第 2.1 で説明したように，本研究では最も簡素な解析手法である文節末から係り先を貪欲に同定していくアルゴリズム [1] を用いる．この解析法を利用する理由としては，(1) 実装が簡単であること，及び (2) 二文節が係る確率が直接得られることである．解析は，文末の文節から一つ手前の文節から，文頭に向けて 1 つずつ係り先を同定していくことになる．この手法では，二文節間の係る確率を計算できるため，文末から文頭に向けてビームサーチを行うことができるが，本研究では最も確率が高いものに係る ( $n = 1$ ) とし，貪欲に選択を行う．

また，文節の係り先候補については，非交差条件を考慮した枝刈りを行う．非交差条件は，図 3 に示すように，係り元文節と係り先文節の間にある文節が，係り先文節を越える係り受けを持つことは無いという仮定である．交差のある文は日本語文において起こり得るものの，交差のない係り受けにときほぐせる場合が多い．交差を持つ文は新聞記事等では殆ど見られないため，非交差条件を仮定して解析を行う．

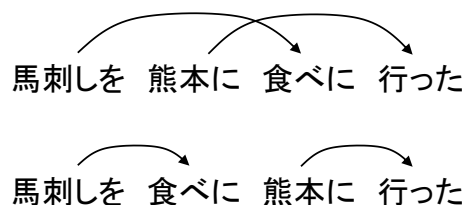


図 3: 交差のある係り受け (上) 及びない係り受け (下)



## 4 既存研究との比較と分析

本章では、はじめに高度な専門性により設計された組み合わせ素性を用いた内本らの手法 [1] と、同じ素性設定で基本素性のみを入力としたニューラルネットとの比較実験を行い、ニューラルネットが人手で獲得されていた組み合わせ素性を獲得することができるかについての検証を行う。また、実際にどのような組み合わせ素性が獲得されたのかについて、学習された入力-中間重み行列  $W_1$  の内部を可視化することにより、分析を行う。

その後、同じ解析法で、多項式カーネルを用いた SVM によるアプローチを取っている工藤らの手法 [2] との比較を行い、多項式カーネルにより暗黙的に組み合わせ素性を獲得した場合に対し、ニューラルネットが同等の性能を出すことができるかについての実験を行う。

### 4.1 内元ら [1] との比較

#### 4.1.1 ベースライン

ベースラインとして、絶対モデルを最大エントロピー法で学習した内本ら [1] のモデルを用いる。内元らは、基本素性に加え、言語的直感に基づいて設計された基本素性の組み合わせ素性を考慮している。しかしながら、組み合わせを人手により列挙するためには、高度な専門性が必要とされ、また人手では有効な組み合わせをすべて列挙しきれないという欠点がある。これに対し、内本らの素性設定において、組み合わせ素性をすべて除き、基本素性のみを用いてニューラルネットを学習し、組み合わせ素性の列挙の必要なしに、より良い組み合わせ素性が獲得されるかの調査を行った。

内本らの組み合わせ素性は、表 1 で示した基本素性の 2～5 つ組を抽出する。具体的な素性については (付録 A) で示す。内元らは、係り元・係り先の二文節間から抽出した基本素性と組み合わせ素性を用いて、ロジスティック回帰 (最大エントロピー法) により二文節に係る確率をモデル化した。

入力文  $\mathbf{x}$  が与えられた時、文中の文節  $d_i$  が文節  $d_j$  に係る確率は、

$$p(d_i \rightarrow d_j | \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \phi(\mathbf{x}, d_i, d_j) - b)} \quad (9)$$

と表される。ここで、 $\mathbf{w}$  は重みベクトル、 $b$  はバイアス、 $\phi$  は素性関数である。素

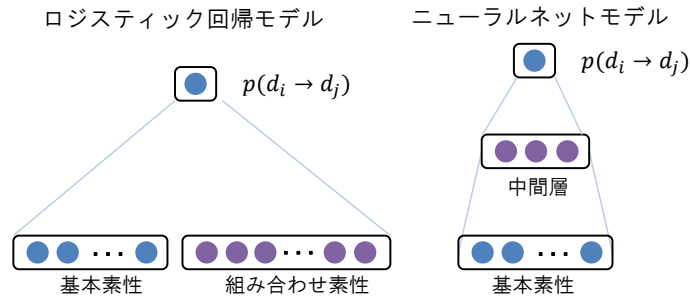


図 4: ロジスティック回帰と 3 層ニューラルネット

性関数  $\phi$  は、基本素性と組み合わせ素性からなる  $0, 1$  ベクトルを返す関数である。ロジスティック回帰は、 $p$  のロジットが線形であると仮定するモデルであり、上式は

$$\log\left(\frac{p}{1-p}\right) = \mathbf{w}^T \phi(\mathbf{x}, d_i, d_j) + b \quad (10)$$

と変形できる。

ロジスティック回帰によるモデルとニューラルネットによるモデルによる差異をわかりやすく表したものを図 4 に示す。大きな違いはロジスティック回帰は入力として明示的に組み合わせ素性（人手による設計）を加えているのに対し、ニューラルネットモデルは入力間の組み合わせを中間層で暗黙的に獲得することを試みる点である。内元らのロジスティック回帰モデルに対し、ニューラルネットモデルの性能が高ければ、人手では捉えきれなかった有益な組み合わせを獲得できているといえる。

## 4.2 使用する基本素性

ニューラルネットへの入力として加える基本素性として使うものは、内本ら [1] の手法で使われた 43 種の素性である。表 1 に基本素性の詳細を示す。これらの基本素性は、多少の差異はあれど、日本語係り受け解析においてほぼ等しく使われているものである。

表 1 において、素性は係り元及び係り先に関する素性、文節間に関する素性、その他の素性に分けられる。係り元及び係り先に関する素性において、主辞と述べられているのは、文節内の形態素のうち、品詞が特殊または助詞であるものを除く最も文末の形態素であり、語形は、品詞が特殊であるものを除く最も文末の形態素である。また、助詞 1 は文節中の最も右端にある助詞を示し、同じく助詞 2 は右端から 2 番めの助詞を表す。

これらの基本的な係り元と係り先の情報に加え、文節間に関する情報も抽出する。文節間距離はバケツ素性であり、文節が隣り合っている場合は A、2～5 文節離れている場合は B、6 文節以上離れている場合は C となる。また、文節間に存在する文節について、係り元と同じ語形を持つものや、係り先と同一の主辞を持つ文節も抽出する。この素性は並列構造を捉えるために有効である考えられている。

表 1: 使用した基本素性

位置	詳細
係り元及び 係り先	主辞表層形・品詞・品詞再分類・活用形・活用型 語形表層形・品詞・品詞再分類・活用形・活用型 句読点有無 助詞 1 表層形・品詞、助詞 2 表層形・品詞 括弧開・閉の有無
文節間	文節間距離 (A: 1, B: 2～5, C: 6～) 括弧開・閉の有無 係り元同一語形・品詞・品詞再分類・活用形・活用型 係り先同一主辞・品詞・品詞再分類・活用形・活用型

### 4.2.1 実験設定

ベースラインで用いられている素性は，組み合わせ素性も含むすべての素性について，出現頻度が3以下のものを除外してある．これにならい，出現頻度が3回以上の基本素性をニューラルネットの入力とした．学習・開発・テストについては，京都大学テキストコーパス Version 4.0<sup>1</sup>を用いた．詳細について表2に示す．

表 2: 学習及びテストに用いたコーパス

学習	1月1日,3日-8日分
開発	1月10日分
テスト	1月9日分

ベースラインは Classias<sup>2</sup> を使い，LBFGS を用いたロジスティック回帰により学習を行った．ニューラルネットは第3章で説明した構成で，Chainer<sup>3</sup> を用いて実装を行った．学習には AdaGrad を使い， $k = 2048$  のミニバッチ学習を適用する．また，中間層の次元を 10, 50, 100, 200, 500 と変化させた場合のニューラルネットを FFNN-10, ..., FFNN-500 と表し，それぞれ学習を行った．モデルのハイパラメータとして，AdaGrad の初期学習率  $\alpha$  及び L2 正則化パラメータ  $\lambda$  があるが，これらは事前に開発セットに対して損失が最小となるようなパラメータのグリッドサーチを行い， $\alpha = 10^{-2}$  及び  $l_2 = 10^{-5}$  を得た．

### 4.2.2 実験結果

実験結果を表3に示す．ここで，係り受け正解率とは，テストセット中に含まれる文末以外の各文節の係り先が正しく推定された割合を示し，文正解率とは文節の係り先がすべて正しく推定されている文の割合を示す．また，表中の「基本素性のみ」とは，基本素性のみを用いてロジスティック回帰を行った場合（組み合わせ素性を一切加えなかった場合）であり，参考までに示した．

実験結果から，ニューラルネットを用いたモデルが，すべての中間層設定において係り受け正解率・文正解率において共にベースラインを越える精度を出して

<sup>1</sup><http://nlp.ist.i.kyoto-u.ac.jp/>

<sup>2</sup><http://www.chokkan.org/software/classias/>

<sup>3</sup><http://chainer.org/>

いることが分かる。また，中間層を増やすことで精度が上昇することが確認できた。これは，中間層が増えるに連れ，より多くの入力間の組み合わせ素性を考慮することができるためであると考えられる。この実験により，人手による列挙では捉えきれなかった有効な組み合わせ素性も，基本素性のみを入力としたニューラルネットであれば獲得することができることが分かった。

表 3: 内本ら [1] の素性設定における係り受け正解率及び文正解率

モデル	係り受け正解率	文正解率
基本素性のみ	70.70	13.93
ベースライン	87.41	44.10
FFNN-50	88.11	46.07
FFNN-100	88.21	45.98
FFNN-200	88.24	46.64
FFNN-500	88.44	45.90

### 4.3 学習された重みパラメータの分析

本節では、ニューラルネットがどのように組み合わせ素性を獲得しているのかを理解するために、学習された重みパラメータの分析を行う。ニューラルネットの組み合わせの獲得は入力-中間層間の線形変換- $\tanh$  関数との間で行われるため、分析を行うパラメータは、入力-中間層の重み行列  $W_1$  である。

学習に用いた素性設定は内元らの素性設定（表1）で、学習するニューラルネットは中間層を50次元とした。各基本素性テンプレートに対し、対応する入力-中間層の重みの絶対値が閾値  $\sigma = 0.2$  を超える場合の素性の数をカウントした。さらに、得られたカウント値を各素性テンプレートに対する素性数で割り、各テンプレートの素性が閾値を超える割合を算出した。これを各中間層毎に繰り返し、素性テンプレートと中間層の発火パターンを視覚的に表したものを図5に示す。

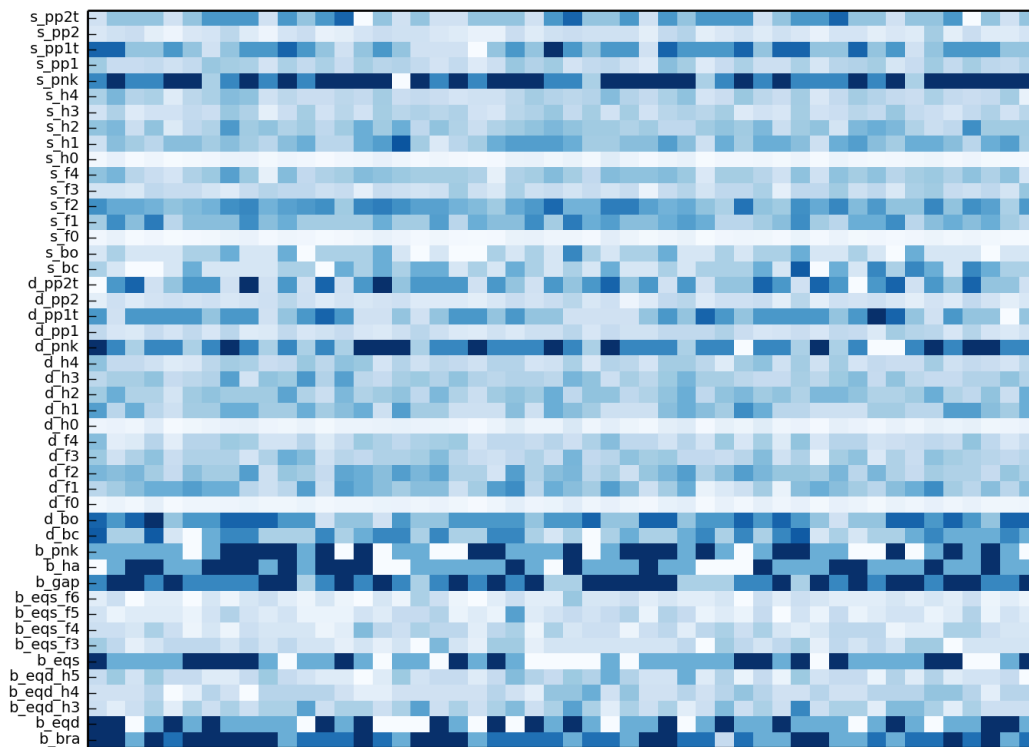


図 5: 入力-中間層行列の発火パターン（縦軸：基本素性テンプレート，横軸：対応する中間層）

図5の縦軸は基本素性テンプレート，横軸は各中間層（50次元）である。色

が濃い部分は、大きな重みを持つ素性の割合が大きいことを示している。図5において、強く参照されやすい基本素性は、係り元読点有無 (s\_pnk), 係り先読点有無 (d\_pnk), 係り先括弧有無 (d\_bo/d\_bc), 文節間句読点 (b\_pnk), 文節間「は」 (b\_ha), 文節間距離 (b\_gap), 係り元同一語形及び係り先同一語形 (b\_eqs/b\_eqd), 文節間括弧 (b\_bra) である。これらは係り受け解析において重要な役割を果たす素性であり、ニューラルネットがそれらを元に組み合わせの獲得を行っていることが分かる。

しかしながら、係り元主辞・語形表層 (s\_h0/s\_f0) 及び係り先主辞・語形表層 (d\_h0/d\_f0) は重みが閾値を超えた素性が殆どない。これは、表層素性が他の素性に比べて出現数が多いためで、閾値を越える重みを持つ素性の割合ではなくカウント値を表したものを図6に示す。この図ではカウント値をそのまま反映しているため、素性数が大きな表層素性 (s\_h0/s\_f0/d\_h0/d\_f0) のみが示される結果となる。

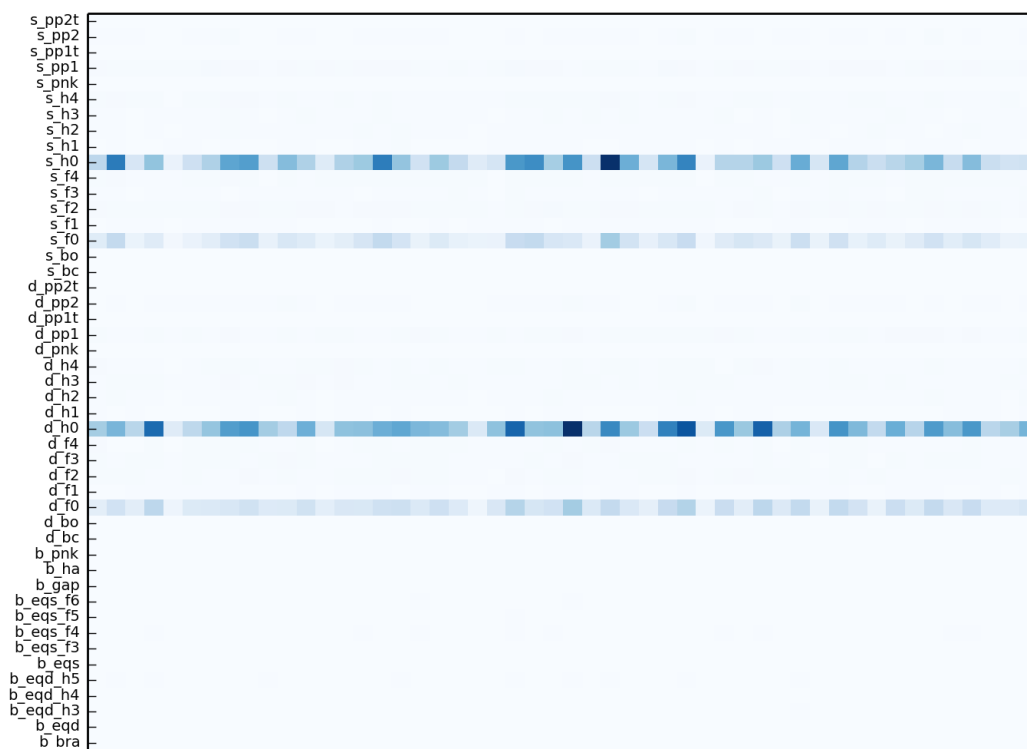


図 6: 入力-中間層行列の発火パターン

図6から、表層形素性には閾値を超える重みが割り当てられる素性が多く存在するが、図5と対比させると、強い重みを持つ素性の割合が、他の素性に比べて圧倒的に少ないことが伺える。即ち、表層素性の内の幾らかは係り受け解析に貢献するような重みの学習が行われるが、それ以外の殆どの素性は解析に貢献しないことになる。このような結果が得られる原因として、文節の係り関係に、係り元と係り先の定型的な表現が大きく影響しており、定型的な表現以外のものは殆ど係り関係に解析に影響を及ぼさないということが考えられる。

## 4.4 工藤ら [2] との比較

### 4.4.1 ベースライン

同じく文末から係り先を決定していく解析法を用いた手法として、工藤らの手法 [2] が挙げられる。工藤らは多項式カーネルを用いた SVM を学習し、学習されたパラメータを用いて擬似的に二文節に係る確率を計算するという、Support Vector Regression [19] に近い手法が用いられている。具体的には、入力及びサポートベクトルから算出したスコアにハイパボリックタンジェント関数を適用し、擬似的な確信度として扱っている。

非線形カーネルを用いた SVM の学習に対する計算量は、学習事例数  $n$  に対して  $O(n^3)$  となる [20]。そのため、学習事例数が大となる係り受け解析では、学習に非常に長い時間を要する。これに対し、ニューラルネットは適切な学習率、正則化重み、パラメータの初期値を用いれば、これらに対して遥かに高速に学習を終えることができる。

工藤らは、計算量の問題に対処するため、学習コーパスを京大コーパス 1 月 1 日～6 日分までに限定している。本節では、実験設定を工藤らと同じ素性設定及び学習データにした上で、非線形カーネルを用いて暗黙的に組み合わせ素性の獲得を行う SVM と比較して、同じく暗黙的に組み合わせを獲得するニューラルネットが、どれだけの性能を示すことができるかの検証を行う。

### 4.4.2 素性設定及び実験設定

工藤らの用いた素性設定は、内本らが用いた素性と殆ど同じだが、動的素性を考慮している点で異なる。動的素性とは、解析時に得られる情報を素性として加えたものであり、解析精度を向上させることが知られている [2][5][4][3][6]。本実験で用いる素性表を 4 に示す。



表 4: 使用した基本素性

位置	詳細
係り元及び 係り先	主辞表層形・品詞・品詞再分類・活用形・活用型 語形表層形・品詞・品詞再分類・活用形・活用型 括弧開・閉の有無 句読点の有無 文頭・文末
文節間	文節間距離 (A: 1, B: 2 ~ 5, C: 6 ~) 括弧の有無 格助詞の有無 句読点の有無
動的素性	係り先に係る文節の語形表層形

ここで述べられている動的素性は、文末から解析を行っていく際に、前の解析ステップで得られた係り情報を元に作成される。また、工藤らと同様に、素性の最低出現頻度による除去は行わず、1度でも出現したすべての素性をモデルへの学習に加えた。

#### 4.4.3 実験結果

工藤らの手法と比較した場合のニューラルネットの結果を表5に示す。実験結果から、多項式カーネルを用いたSVMに対し、同じコーパス及び素性設定を用いたニューラルネットが近い性能を示すことが分かった。これは、多項式カーネルを用いたSVMが暗黙的に獲得していた組み合わせ素性と同等の性能を示す組み合わせを、ニューラルネットが獲得できる事を示唆している。

ニューラルネットを学習は、CPU: Xeon E5-2430 v2(2.5GHz)において、約2時間を要した。但し実装はPythonにより行っているため、最適化のかかる言語による実装を行えばさらなる高速化が望める。また、ニューラルネットを用いた場合には、行列計算に粗演算を用いること [21] や、予めよく出現する入力素性についての行列演算を行っておくという工夫 [13] が行えるため、更に高速な学習及び解析が可能である。

ニューラルネットの学習及び解析は、多項式カーネルを用いた場合のSVMに

比べて遥かに速いことを考えると、より短い学習時間で効率的に組み合わせを獲得できているということが言える。

表 5: 工藤ら [2] の素性設定における係り受け正解率及び文正解率

モデル	係り受け正解率	文正解率
FFNN-500	87.98	44.10
工藤ら [2]	88.66	44.47

## 5 単語分散表現の導入の検討

本節では、まずはじめに単語分散表現についての解説を行い、その後、ニューラルネットを用いた日本語係り受けへの単語分散表現導入の検討を行う。

### 5.1 単語分散表現とは

単語分散表現とは、単語素性の one-hot ベクトル（第 3.3 節参照）の代わりに、単語一つ一つに密なベクトルを割り当てることにより単語を表現すること指す。この時に用いられる密なベクトルは、例えば単語共起頻度行列に対して特異値分解（SVD）による次元圧縮を行ったものなどが用いられる。分布仮説によれば、似たような意味を持つ単語同士は同じような共起パターンを持つ。従って、単語共起行列を圧縮することによって得られる分散表現は、共起パターンにおけるノイズ成分を減らしつつ、似た意味の単語は似たベクトルを持つような表現となる。

入力として one-hot ベクトルの代わりに単語類似度が正しく反映されるような単語分散表現を用いることは、一般的に有益であると考えられている。例えばあるタスクにおいて、学習に使用できるデータが限られており、学習時に出現しなかった単語がテスト時に未知語として出現したとする。この時、テスト時では未知語に対して「未知語が来た」という素性を入力する程度のことしかできず、十分に対処することができない。しかしながら、単語分散表現は主タスクとは別の大規模コーパスにより学習し、大きなボキャブラリを持っているため、主タスクにおいて未知語であったとしても、未知語の単語分散表現を用いることで、主タスク中に存在する似た意味の別の単語と同じように扱われることができる。

より良い単語分散表現を得るために、これまでに様々な手法が提案されてきたが、Mikolov らが提案した word2vec[22][23] は、Noise Contrastive Estimation[24][25] を用いることより学習時間を大幅に短縮しつつ、単語類似度と単語アナロジーをよく表す単語分散表現の獲得ができるとして注目を浴びた。

近年、大規模コーパスから学習済みの単語分散表現をモデルの入力として加えるだけでなく、それらを初期値とするパラメータを用意し、主タスクでの学習時に更新することにより、より良い性能を出すとする研究が報告されている [12].

## 5.2 英語における類似モデル

英語における係り受けタスクでは，Chen らが3層のニューラルネットを用いた手法を提案しており [13]，既存手法と並ぶ精度を達成しつつ，より高速な解析を行ってみせた．それに続くように，より複雑な構成のニューラルネットを用いて係り受けタスクを解くモデルが次々と提案され，state-of-the-art の性能を競っている [14][15]．しかしながら，これらの手法ではニューラルネットに加え，例外なく単語分散表現も使用されている．単語分散表現を使うメリットは上記第5.1で説明したが，各種法が達成した成果がニューラルネットによるものか，単語分散表現によるものかが明らかではなかった．

単語分散表現のメリットやChen らの先行研究から考えれば，同じくニューラルネットを用いた日本語の係り受け解析において，単語分散表現を用いることによる性能向上が望めるはずである．本章では以降，第3章及び第4章で述べてきた日本語係り受け解析に対するニューラルネットに対し，単語分散表現を適用した場合の影響について詳細に議論する．

### 5.3 ニューラルネットに基づく日本語係り受け解析への単語分散表現の導入

本節では、単語分散表現の有効性を検証するために、単語分散表現を用いた場合と用いなかった場合のニューラルネットの比較実験を行う。ベースラインとするモデルは第3章及び第4章で説明した中間層200次元のニューラルネットである。このニューラルネットに対し、係り（元・先）文節の（主辞・語形）の表層形素性を、one-hotベクトルから単語分散表現に変更した場合についての実験を行う。主辞・語形の表層形素性のみに対して単語分散表現を適用する理由としては、素性の異なり数が少ないような素性（例：文節間距離（3値））はテストデータ中で未知語となる確率が低いため、分散表現を用いても効果がないと考えられるためである。

表6に、テストデータ中に出現するが学習データに出現しない未知語数を各表層素性について求めたものを示す。テストセットで出現する表層素性のうち、約30%程度が未知語であり、テストセット中に多くの未知語が存在することが分かる。よって、大規模コーパスから予め未知語に対する分散表現を学習することにより、未知語への対応関係が捉えられ、性能向上が望めると考えられる。

表 6: テストデータにおける各表層素性の未知語数（未知語数/素性数）

係り元主辞	係り元語形	係り先主辞	係り先語形
1325/4396	566/1636	1357/4472	636/1855

#### 5.3.1 実験設定

素性設定は第4.1節で用いた、内本らの基本素性（表1）を用いる。係り元主辞表層形・係り元語形表層形・係り先主辞表層形・係り先語形表層形について、それぞれ独立に50次元の単語分散表現を用意する。単語分散表現の学習は、Mikolovらのword2vec[22][23]<sup>4</sup>を用い、日本語Wikipedia（約1.9GB）により行った。中間層はChenらに従い、200次元とした。また、大規模コーパスで学習した単語分散表現の初期化によるメリットを見るために、ランダムに初期化した分散表現を用いたモデルも用意した。

<sup>4</sup><https://code.google.com/archive/p/word2vec>

### 5.3.2 実験結果

得られた実験結果を7に示す。実験結果から、単語分散表現を導入した場合は、導入しなかった場合に比べて精度が下がる結果となった。また、単語分散表現の初期化を word2vec を用いて行った場合の精度は、ランダムに初期化した場合を下回る結果となった。

表 7: 単語分散表現の初期化に対する精度の変化

単語分散表現	初期化	係り受け正解率	文正解率
無し		88.65	47.05
50次元	word2vec	87.87	43.85
50次元	ランダム	88.19	44.67

まず、ランダムに初期化した場合が word2vec で初期化した場合に比べて精度が高いことについての考察を行う。ランダムによる単語分散表現行列は、中間層の各ノードの値が平均0、分散1となるように初期化される。このように初期化しなかった場合、中間層のハイパボリックタンジェントへの入力小さすぎる、あるいは大きすぎるため、学習に非常に時間がかかったり、局所解に陥りやすくなる事が経験的に知られている [17]。しかしながら、大規模コーパスにより学習された単語分散表現は、初期値とする単語分散表現行列の平均と分散が正しく設定されているとは限らない。また、二つの単語の類似度が単語ベクトルのコサイン類似度<sup>5</sup>により表わされるように学習されているため、分散表現の行列に対して、次元毎の正規化を行うことができない。

次に、単語分散表現を用いない場合に比べて精度が悪化していることについての考察を行う。単語分散表現を用いない場合は、入力層には単語表層形 one-hot ベクトルが加えられる。しかしながら、単語分散表現を用いた場合は、one-hot ベクトルに分散表現の行列をかけて単語分散表現を得て、入力として加える事になる。即ち単語分散表現を取り入れることは、one-hot ベクトルを入力とした際の入力-中間層の間に、単語分散表現のための行列を組み込む操作と等価であり、一種の行列分解を行っている事になる。この時の分散表現の次元数  $d$  はモデルのハイパパラメータとなり、最適な  $d$  が存在すると考えられる。

<sup>5</sup>コサイン類似度  $\text{cossim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{|\mathbf{x}| |\mathbf{y}|}$

最適な  $d$  を用いた単語分散表現が、単語分散表現を用いなかった場合に比べてどのような結果となるかについて次節で考察を行う。

## 5.4 次元数を変化させた場合

単語分散表現の次元数  $d$  を調節することで、少ないパラメータで元の行列を表すような一種の正則化の効果が得られるため、最適な  $d$  が存在すると考えられる。第 5.3.1 節における実験において、分散表現の次元は Chen らのモデル [13] に従い 50 次元としたが、この分散表現の次元数  $d = 10, 50, 100, 150, 200$  を変化させた場合に、精度にどのような変化が見られるかの実験を行った。また、word2vec による単語分散表現による初期化による精度向上は見られなかったため、ここでは単語分散表現の初期値をランダムとしている。実験結果を表 8 に結果を示す。

表 8 から、次元  $d$  が中間層の次元より小さい場合、入力-中間層重み行列の行列分解を行うことになる。以下の場合、 $d = 100$  の時にもっともよい精度を出すことができた。

しかしながら、単語分散表現を用いなかった場合（分散表現次元数＝なし）に比べ、精度は低い結果となった。これは、分散表現を用いない場合の入力-中間層の重み行列を分解することが、精度に悪影響を及ぼしているということを示している。このことから、第 4.3 節で説明したように一部の表層素性のみが強く参照され、それ以外の素性は余り出力値に影響するように学習されないような場合、分散表現の導入による行列分解を行わずに、直接素性の one-hot ベクトルを入力として加えた方が良いということが考えられる。

表 8: 単語分散表現の次元数を変化させた場合

中間層次元数	分散表現次元数	係り受け正解率	文正解率
200	なし	88.65	47.05
200	10	87.79	44.01
200	50	88.19	44.67
200	100	88.34	45.49
200	150	88.18	45.00
200	200	87.98	45.08

## 6 おわりに

本研究では、日本語係り受け解析において、文末から貪欲に同定していく解析法に対する3層フィードフォワードニューラルネットの実装を行い、その有用性を証明するために、既存手法の持つ欠点を克服することができるかについて、実証を行ってきた。まずはじめに、高度な専門性により人手で設計された組み合わせ素性によるモデルに対し、基本素性のみを入力とした場合のニューラルネットが、人手では捉えきれなかった組み合わせ素性の獲得により、より高い精度を達成できることを証明した。またその際に、学習された入力-中間層重み行列を可視化することで、どのような組み合わせが獲得できているかを把握でき、係り受け解析において有用だと考えられる素性が抽出されていることが分かった。

その後、ニューラルネットと同じく、入力間の組み合わせを暗黙的に考慮できるモデルとして、多項式カーネルを使ったSVMとの比較実験を行った。実験から、ニューラルネットは多項式カーネルを用いたSVMに比べて僅かに低い精度となるものの、SVMより高速に学習・解析ができる事を示した。

最後に、本研究の後半において、海外の係り受けタスクにおいて本研究と同じモデルを用いたChenらの研究にならい、日本語係り受け解析に対して単語分散表現を取り入れた場合について各種実験を行った。大規模コーパスにより学習した単語分散表現を初期値とした実験から、テストセットに多くの未知語が有るにも関わらず、精度を上げることはできなかった。しかしながら、単語分散表現の初期値をランダムとした上で、分散表現の次元数がモデルに与える影響について、行列分解という観点からの説明を行うことができた。



## 謝辞

本研究を進めるにあたり，ご協力，御助言を頂きました多くの方々に，深く感謝いたします。主指導教員である乾健太郎教授には，研究の進め方やどうあるべきかについて繰り返しご指導いただき，深く感謝いたします。また，関連手法についてのご助言や研究方針についてアドバイスを下さった岡崎直観准教授に心より感謝いたします。審査委員をお引き受け下さいました，木下賢吾教授，伊藤彰則教授に深く感謝します。詳細な手法に関する的確なアドバイスを下さった田然研究特任助教や，お忙しい中で温かいご指導を下さいました井之上直也助教に心から感謝申し上げます。最後になりましたが，研究生生活の様々な場面でお世話になりました研究室の皆様にも，感謝を申し上げます。

## References

- [1] 内元清貴, 関根聡, 井佐原均. 最大エントロピー法に基づくモデルを用いた日本語係り受け解析. 情報処理学会論文誌, 40 (9), pp. 3397-3407, 1999.
- [2] 工藤拓, 松本裕治. Support Vector Machine による日本語係り受け解析. 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション, 100 (201), pp. 25-32, 2000.
- [3] 颯々野学. 日本語係り受け解析の線形時間アルゴリズム, 自然言語処理, 14 (1), pp. 3-18, 2007.
- [4] 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, 43 (6), pp. 1834-1842, 2002.
- [5] 工藤拓, 松本裕治. 相対的な係りやすさを考慮した日本語係り受け解析モデル. 情報処理学会論文誌, 46 (4), pp. 1082-1092, 2005.
- [6] 岩立将和, 浅原正幸, 松本裕治. トーナメントモデルを用いた日本語係り受け解析. 自然言語処理, 15 (5), pp. 169-185, 2008.
- [7] Taku Kudo and Yuji Matsumoto. Fast methods for kernel-based text analysis. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Volume 1, pp. 2431, 2003.
- [8] Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Haji. Non-projective dependency parsing using spanning tree algorithms. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 523-530, 2005.
- [9] Yoav Goldberg and Michael Elhadad. splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, pp. 237-240, 2008.
- [10] Naoki Yoshinaga and Masaru Kitsuregawa. Polynomial to linear: efficient classification with conjunctive features, In Proceedings of the Conference on

Empirical Methods in Natural Language Processing, Volume 3, pp. 1542-1551, 2009.

- [11] Naoki Yoshinaga and Masaru Kitsuregawa. Kernel Slicing: Scalable Online Training with Conjunctive Features, In Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1245-1253, 2010.
- [12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, Volume 12, pp. 2493-2537, 2011.
- [13] Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 740-750, 2014.
- [14] Hao Zhou, Yue Zhang, Shujian Huang, Jiajun Chen. A Neural Probabilistic Structured-Prediction Model for Transition-Based Dependency Parsing. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015.
- [15] Wenzhe Pei, Tao Ge, Baobao Chang. An Effective Neural Network Model for Graph-based Dependency Parsing, In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015.
- [16] 熊沢逸夫. 電子情報通信工学シリーズ 学習とニューラルネットワーク. 森北出版株式会社, pp. 21-33. 2007.
- [17] Yann LeCun, Leon Bottou, Genevieve B. Orr, Klaus Robert Mller. Efficient BackProp. Neural Networks: Tricks of the Trade, 2nd edn. ,pp. 9-48. 2012.
- [18] John Duchi, Elad Hazan, Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. The Journal of Machine Learning Research, Volume 12, pp. 2121-2159, 2011.
- [19] Alex J. Smola and Bernhard Schlkopf. A tutorial on support vector regression. Statistics and Computing, Volume 14, pp. 199-222, 2004.

- [20] Lon Bottou, Chih-jen Lin. Support Vector Machine Solvers, Large Scale Kernel Machines, pp. 1-28. 2007.
- [21] Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. *Neural Networks: Tricks of the Trade*, 2nd edn., pp. 437-478. 2012.
- [22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*, 2013.
- [23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Neural Information Processing Systems 26*, 2013.
- [24] Michael U. Gutmann, Aapo Hyvriinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Volume 9, pp. 297-304, 2010.
- [25] Michael U. Gutmann, Aapo Hyvriinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. In *Journal of Machine Learning Research*, Volume 13, pp. 307-361, 2012.

## 付録

### A 内本 [1] における素性設定

表 9: 基本素性一覧

ID	素性名	素性数	例
	係り元素性		
0	主辞表層形	13991	して, する, した, し, こと
1	主辞品詞	14	名詞, 動詞, 形容詞, 副詞, 指示詞
2	主辞品詞細分類	22	普通名詞, サ変名詞, 数詞, 時相名詞
3	主辞活用形	31	サ変動詞, 母音動詞, 子音動詞ラ行, ナ形容詞
4	主辞活用形細分類	42	基本形, タ系連用テ形, 基本連用形, タ形
5	語形表層形	5544	の, は, を, に, が
6	語形品詞	14	助詞, 動詞, 名詞, 接尾辞, 形容詞
7	語形品詞細分類	27	格助詞, 接続助詞, 副助詞, 普通名詞
8	語形活用形	33	母音動詞, サ変動詞, 子音動詞ラ行, ナ形容詞
9	語形活用形細分類	39	基本形, 基本連用形, タ形, タ系連用テ形
10	助詞 1	60	の, は, を, に
11	助詞 1 品詞	5	格助詞, 接続助詞, 副助詞, 終助詞
12	助詞 2	44	で, に, など, と
13	助詞 2 品詞	5	格助詞, 副助詞, 接続助詞, 終助詞
14	句読点有無	3	、, 。, None
15	括弧開の有無	6	「, 『, (, “, None
16	括弧閉の有無	6	」, 』, ) , ”, None
	係り先素性		
17	主辞表層形	14271	した, して, する, こと, さ
18	主辞品詞	14	名詞, 動詞, 形容詞, 判定詞, 助動詞
19	主辞品詞細分類	23	普通名詞, サ変名詞, 数詞, 形式名詞
20	主辞活用形	32	サ変動詞, 母音動詞, 子音動詞ラ行, 子音動詞ワ行
21	主辞活用形細分類	41	基本形, タ形, タ系連用テ形, 基本連用形
22	語形表層形	6345	と, が, の, を, に
23	語形品詞	14	助詞, 動詞, 接尾辞, 名詞, 形容詞

ID	素性名	素性数	例
24	語形品詞細分類	28	格助詞, 動詞性接尾辞, 接続助詞, 副助詞
25	語形活用形	34	母音動詞, サ変動詞, 子音動詞ラ行, 子音動詞ワ行
26	語形活用形細分類	39	基本形, タ形, 基本連用形, タ系連用テ形
27	助詞 1	64	と, が, の, を
28	助詞 1 品詞	5	格助詞, 接続助詞, 副助詞, 終助詞
29	助詞 2	41	と, など, に, で
30	助詞 2 品詞	5	格助詞, 副助詞, 接続助詞, 終助詞
31	句読点有無	3	。 ,、 ,None
32	括弧開の有無	6	「, “, 『, (, None
33	括弧閉の有無	6	」, 』, ”, ) ,None
34	文節簡素性 距離	3	A,B,C
35	読点の有無	2	True,False
36	「は」の有無	2	True,False
37	括弧開閉の有無	4	NN,OC,NC,ON
38	係り元同一語形文節 有無	2	False,True
39	主辞品詞	12	助詞, 動詞, 接尾辞, 名詞
40	主辞品詞細分類	21	格助詞, 接続助詞, 副助詞
41	主辞活用形	18	サ変動詞, 母音動詞, イ形容詞アウオ段
42	主辞活用形細分類	20	基本形, タ形, 基本連用形
43	係り先同一主辞 有無	2	False,True
44	語形表層形	9	動詞, 名詞, 判定詞, 助動詞
45	語形品詞	12	普通名詞, 形式名詞, サ変名詞
46	語形品詞細分類	17	サ変動詞, 判定詞, 子音動詞ラ行

表 10: 組み合わせテンプレート一覧

	組み合わせテンプレート (ID 表記)
2 素性	(5,17), (6,17), (7,17), (5,18), (6,18), (7,18), (5,19), (6,19), (7,19)
3 素性	(5,18,34), (6,18,34), (7,18,34), (5,19,34), (6,19,34), (7,19,34), (5,18,35), (6,18,35), (7,18,35), (5,19,35), (6,19,35), (7,19,35), (5,18,36), (6,18,36), (7,18,36), (5,19,36), (6,19,36), (7,19,36), (5,18,37), (6,18,37), (7,18,37), (5,19,37), (6,19,37), (7,19,37), (5,18,38), (6,18,38), (7,18,38), (5,19,38), (6,19,38), (7,19,38), (5,18,39), (6,18,39), (7,18,39), (5,19,39), (6,19,39), (7,19,39), (5,18,40), (6,18,40), (7,18,40), (5,19,40), (6,19,40), (7,19,40), (5,18,41), (6,18,41), (7,18,41), (5,19,41), (6,19,41), (7,19,41), (5,18,42), (6,18,42), (7,18,42), (5,19,42), (6,19,42), (7,19,42), (5,18,43), (6,18,43), (7,18,43), (5,19,43), (6,19,43), (7,19,43), (5,18,44), (6,18,44), (7,18,44), (5,19,44), (6,19,44), (7,19,44), (5,18,45), (6,18,45), (7,18,45), (5,19,45), (6,19,45), (7,19,45), (5,18,46), (6,18,46), (7,18,46), (5,19,46), (6,19,46), (7,19,46), (32,33,37), (10,12,18), (10,12,19), (11,13,18), (11,13,19)
4 素性	(5,18,14,31), (6,18,14,31), (7,18,14,31), (5,19,14,31), (6,19,14,31), (7,19,14,31), (0,5,17,22), (0,6,17,23), (0,7,17,24), (1,5,18,22), (1,6,18,23), (1,7,18,24), (2,5,19,22), (2,6,19,23), (2,7,19,24), (5,18,38,43), (6,18,38,43), (7,18,38,43), (5,19,38,43), (6,19,38,43), (7,19,38,43)
5 素性	(0,5,17,22,34), (0,6,17,23,34), (0,7,17,24,34), (1,5,18,22,34), (1,6,18,23,34), (1,7,18,24,34), (2,5,19,22,34), (2,6,19,23,34), (2,7,19,24,34), (1,10,12,18,22), (1,11,13,18,22), (2,10,12,19,22), (2,11,13,19,22), (1,10,12,18,23), (1,11,13,18,23), (2,10,12,19,23), (2,11,13,19,23), (1,10,12,18,24), (1,11,13,18,24), (2,10,12,19,24), (2,11,13,19,24)