原著論文

# ヒューマノイドロボットを用いた言語理解による動作生成†

濵園　侑美*1・小林　一郎*2・麻生　英樹*3・中村　友昭*4・長井　隆行*5・持橋　大地*6

現在，日本は超高齢化社会に突入しており，それに伴う人手不足問題をロボットを利用して解決をはかろうとする取り組みが増えている．近年では，ロボットを安価に入手出来るようになったり，人とロボットが触れ合う機会も増えており，人間とロボットのコミュニケーションが大きく発展する可能性がある．本研究は，家庭内でロボットを用いる場面を想定し，人の言葉を理解し動作を生成するヒューマノイドロボットの実現に向けた取り組みである．言葉の意味を捉える方法は，単語が元々持っている意味や単語同士の意味的関係から意味ネットワークを生成するも方法があるが，本研究では，言葉の意味の捉え方に文章中の単語の頻度情報や周辺に出現する単語の類似を用いて自動的に意味を付与する手法である，分散表現を用いる．それにより，曖昧な言語表現であっても分散表現上では一つの表現に定まり，扱いが容易になる．ロボットに動作を生成させる方法として，ロボットの関節の動きと人間の関節の動きは異なるということに留意し，ロボットが行える基本動作を組み合わせることにより，人の動きを真似て行える枠組みを提案する．この枠組みにより，基本動作を組み合わせて，複雑な動作を表現および生成すること可能がとなる．言葉と動作の対応関係を捉える方法は様々あるが，本研究ではニューラルネットワークの技術を用い，多様なロボット動作と曖昧な表現との対応関係を学習する枠組みを提案し検証する．これらにより，人の言葉による指示からロボットが動作を生成し動作することが可能となる．また，初めて行う動作であっても言葉の意味から推測し，ロボットが動作生成することが可能となる．

キーワード：自然言語処理，機械学習，ヒューマノイドロボット，ニューラルネットワーク，マルチタスクラーニング

## 1. はじめに

少子高齢化社会の到来によって起こりうる人手不足の問題を，ロボット利用によって解決をはかろうとする場面が増えると考えられるいま，家庭内において人間とロボットが協調して暮らせるように，ロボットは居住者の言葉や身振りをまねることで居住者の経験を得て，学習することが必要となると考える．このことを踏まえ，本研究は人の言葉による指示からロボットが動作を生成し動作することを目標にする．言葉は，人間とロボットのインタラクションが容易になるよう，分散表現を用いて表現する．分散表現は多様な意味を持つことがある単語が 1 つの表現に定まり，さらに単語同士の関係が人の想定する意味関係と近い相関関係を取る特徴を持つ．動作は，ロボットが人の動作を真似て行うことができるように，自らが行える基本動作を組み合わせてひとつの行為を生成する枠組みを構築し，複数の基本動作から複雑な行為を表現する対応関係を明確にする．また，言葉と動作の対応関係を学習することによって，初めて行う動作であっても言葉から推測し，動作を行なえるようにし，多様なロボット動作と多様な人の言葉の対応関係を学習する枠組みを提案し検証する．なお，本研究は家庭内でのロボット利用を目指すため，日常生活の動作，特に調理動作を対象とし，言葉は動作を表す動詞や物の名前を表す名詞ではなく，その量や質を表現する副詞や形容詞に注目し，ロボット動作と言葉の対応関係を学習する．

## 2. 関連研究

ニューラルネットワークやリカレントニューラルネットワーク（RNN）を用い，ロボットに動作を生成させる研究は広く行われている．神経力学モデル MTRNN（Multiple Timescale Recurrent Neural Network）[1] は RNN を拡張したモデルで，再起結合を持つことで力学モデルとしての性質を持つ．現在の状態を入力とし，次時刻の動作を出力とするものであり，動作を学習，認知，生成することができる．これを用いて，Sugita ら [2] 及び Ogata ら [3] はパラメータを共有する MTRNN と RNN を用い，ロボット動作と単語列をそれぞれ学習することで，相互想起を実現するモデルを提案した．また，Stramandinoli ら [4] は既に獲得している低次元における動作と言葉の対応関係に基づいて，RNN を用いることでより高次元の動作と言語表現の統合を目指した．しかし，いずれの場合も名詞や動

† Motion Generation Using Humanoid Robot with Language Understanding
Yumi HAMAZONO, Ichiro KOBAYASHI, Hideki ASOH, Tomoaki NAKAMURA, Takayuki NAGAI, and Daichi MOCHIHASHI

*1 お茶の水女子大学大学院 人間文化創成科学研究科
Humanities and Sciences Advanced, Ochanomizu University
*2 お茶の水女子大学大学院 基幹研究院 自然科学系
Natural Science Division, Faculty of Core Research, Ochanomizu University
*3 産業技術総合研究所 人工知能研究センター
Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology(AIST)
*4 電気通信大学
The University of Electro-Communications
*5 大阪大学
Osaka University
*6 統計数理研究所 数理・推論研究系
Department of Statistical Inference and Mathematics, The Institute of Statistical Mathematics

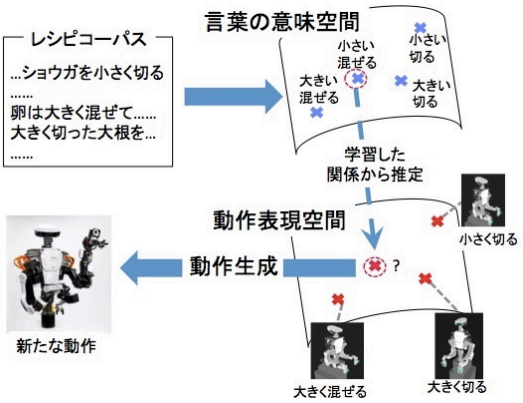



図 1　提案手法の概要（未知語「小さく 混ぜる」の動作推定例）

詞などの具体的な言葉に対する動作生成である．本研究では，より人間の言葉に対応するために，名詞で表される物や動作の大きさや量を具体的な数値として表さず，曖昧に表現する形容詞や副詞の意味理解に基づくロボットの動作生成を行う．

自然言語処理分野においても，ニューラルネットワークを用いて機械翻訳 [5, 6]，文書分類 [7]，固有名詞抽出 [8]，質疑応答 [9] などの様々なタスクを解決する取り組みが行われている．例えば GNMT（Google's Neural Machine Translation）[10] は，Attention 機構 [11] を導入した Encoder-Decoder [6, 12] モデルで，Encoder，Decoder 共に 8 層の LSTM [13] を使用した機械翻訳である．このように，自然言語処理におけるニューラルネットワークの使用の多くは巨大なネットワークの End-to-End である．本研究では，巨大なネットワークを作ることなく学習させるための工夫を行う．

なお，語義曖昧性解消への取り組みとして，Knowledge-Base に基づく手法 [14] や教師あり学習 [15]，教師なし学習 [16]，半教師あり学習 [17] による手法が取られている．近年では，分散表現を用いて解決する手法 [18, 19] が台頭している．しかし，分散表現によって生成された言葉の意味空間はコーパスに依存しており，言葉以外の関係性との乖離が見られる．そこで本論文では，言葉の意味空間と，言葉以外の空間として動作表現空間を結びつける手法を提案する．

## 3. 言語表現からの動作生成

### 3.1 概要

動作を変化させる曖昧表現と動作を意味する動作表現とを組み合わせた言葉と，その言葉が意味する動作の対応関係が既知であるとする．動作との対応関係が分からない未知の言葉が与えられた時に，既知の言葉との意味関係から対応する動作を生成する手法を提案する．

図 1 に提案手法の概要を例と共に示す．

言葉は言葉の意味空間に，動作は動作表現空間にそれぞれ配置され，言葉と動作の関係により 2 つの空間の関係性を学習することで，言葉から動作を推定し，動作が生成できるようになる．

例えば，「大きく 混ぜる」，「大きく 切る」，「小さく 切る」の 3 つの言葉とそれぞれの動作が既知であるとする．この時，未知の「小さく 混ぜる」動作を生成しようとする場合，既知の「大きく 切る」，「小さく 切る」の関係から「大きい」と「小さい」の関係性を推定でき，「大きく 混ぜる」動作と「大きい」と「小さい」の関係性を用いて「小さく 混ぜる」動作を推定し，動作を生成する．

言葉を意味空間へ配置する方法は word2vec [20, 21] を用い，対象とする言葉は，動作を変化させる曖昧表現（「大きい」，「小さい」，「しっかり」，「手早い」などの形容詞や副詞）と，動作に関する単語（「切る」，「混ぜる」，「振る」などの動詞）を組み合わせたものとする．動作を表現空間へ配置する方法は，Activity-Attribute Matrix を動作生成に応用した時系列対応 AAM を用いる．また，言葉と動作の対応関係の学習にはニューラルネットワークを用いる．

### 3.2 言語の意味表現

Hinton ら [22] によって提案された分散表現とは，単語を固定長の高次元の実数ベクトルで表現し，類似の意味を持つ単語を近いベクトルに対応させる手法である．各概念に一つの計算要素を割り当てる局所表現（one-hot representation）に比べ，より単語同士の関係とベクトル同士の関係が対応づけられる．分散表現生成手法のうち，本研究では Mikolov ら [20, 21] によって公開された word2vec[1] を用いる．これはニューラルネットワークを用いて単語の分散表現を得る手法であり，文章中の単語の出現頻度と周辺に出現する単語の類似により，単語をベクトル化する．word2vec は，学習方法の違いにより CBOW と Skip-gram の 2 つに分類されるが，双方とも生成されたベクトル同士の計算によって単語の意味を表現可能である [23] と知られている．これにより，単語の意味関係から未知の単語の組み合わせに対する動作の推定を可能にする．word2vec により分散表現を生成した場合，用いるコーパスに依存して生成されたベクトル同士の関係性が変化する．本研究は調理動作を対象とするため，コーパスはクックパッド[2] のレシピを用いた．また，word2vec を用いる際の前処理としてクックパッドデータの材料コーパスを辞書とすることで，MeCab [24] による分かちを正確にし，既存辞書を用いる場合よりも更に料理に適した分散表現を生成出来るように工夫した．4.1 節で述べる本研究で用いる曖昧表現の word2vec による言葉の分散表現を主成分分析し，可視化した結果が図 2 である．

図 2 より，速度が速いことを表す「手早い」と「一気に」，少量であることを表す「少し」と「ちょっと」がそれぞれ近いことがわかる．一方で，逆の意味である「細かい」と「ザクザク」が近い．これは，word2vec による分散表現の作成前提である，分布仮説 [25] の「同じ文脈に現れる単語は似た意味を持つ傾向にある」により，対義語同士もまた同じ文脈に出やすいため，類義語と対義語は区別することができないことが原因である．これを解消するために，潜在的意味解析をベースとした辞書知識を利用する手法 [26] や，意味的に関連している単語群は似たベクトルになるようにファインチューニングする，

1　https://code.google.com/p/word2vec/
2　http://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html

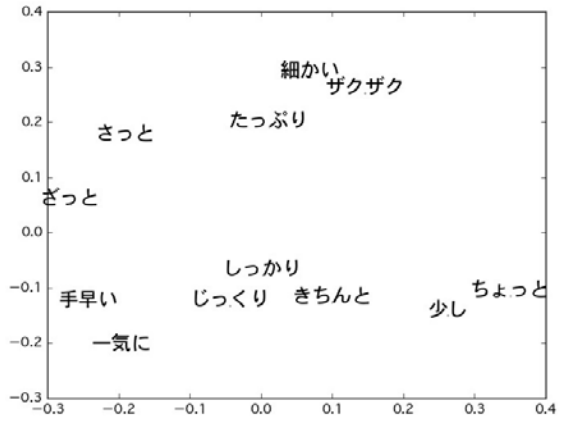



図２　word2vec による分散表現

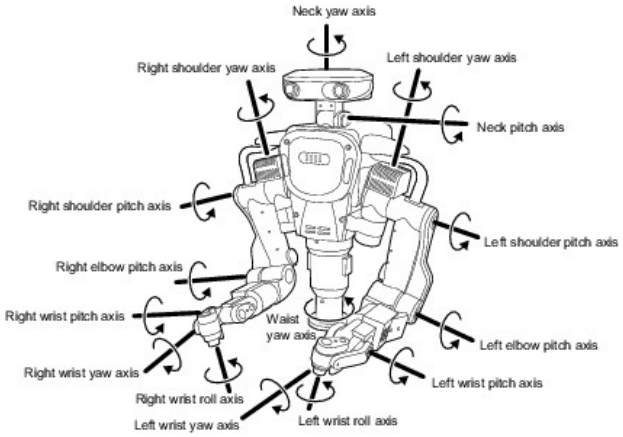



図３　ロボットの関節

Retrogitting という手法 [27] があるが，本研究は分散表現の精度をあげることが目的ではないため，用いない．

### 3.3　ヒューマノイドロボット概観

図 3 に本研究で使用する HIRONXC の概観[3]を示す．

HIRONXC は（株）川田工業社製ヒューマノイドロボットであり，オープンソフトウェア OpenHRP3 を持つプラットフォームである．双碗，双眼を持ち，左右手元にカメラがついて，Jython で動作する．また，全部で 23 つの関節を持っており，体全体を動かす 1 つ（腰ヨー軸：CY），首の左右上下運動に 2 つ（首ヨー軸：NY，首ピッチ軸：NP），両腕に各 6 つ（右肩ヨー軸：RSY，右肩ピッチ軸：RSP．右肘ピッチ軸：REP，右手首ヨー軸：RWY，右手首ピッチ軸：RWP，右手首ロール軸：RWR，左肩ヨー軸：LSY，左肩ピッチ軸：LSP，左肘ピッチ軸：LEP，左手首ヨー軸：LWY，左手首ピッチ軸：LWP，左手首ロール軸：LWR），両手に各 4 つ（右手関節：RH1，RH2，RH3，RH4，左手関節：LH1，LH2，LH3，LH4）の関節を持つ．それぞれの関節角と時間 $T$ を指定することで，$T$ 秒かけて指定された角度へと関節を動かすことが可能である．

3　http://robot-support.kawada.jp/support/hiro/ より引用

表１　関節の動作可能範囲

| 関節軸名称 | 記号 | 可動範囲（deg） | 最大速度（deg/s） |
|---|---|---|---|
| 腰ヨー軸 | CY | -163 to +163 | 130 |
| 首ヨー軸 | NY | -70 to +70 | 150 |
| 首ピッチ軸 | NP | -20 to +70 | 300 |
| 右肩ヨー軸 | RSY | -88 to +88 | 172 |
| 右肩ピッチ軸 | RSP | -140 to +60 | 133 |
| 右肘ピッチ軸 | REP | -158 to +0 | 229 |
| 右手首ヨー軸 | RWY | -165 to +105 | 300 |
| 右手首ピッチ軸 | RWP | -100 to +100 | 223 |
| 右手首ロール軸 | RWR | -163 to +163 | 300 |
| 左肩ヨー軸 | LSY | -88 to +88 | 172 |
| 左肩ピッチ軸 | LSP | -140 to +60 | 133 |
| 左肘ピッチ軸 | LEP | -158 to +0 | 229 |
| 左手首ヨー軸 | LWY | -105 to +165 | 300 |
| 左手首ピッチ軸 | LWP | -100 to +100 | 223 |
| 左手首ロール軸 | LWR | -163 to +163 | 300 |

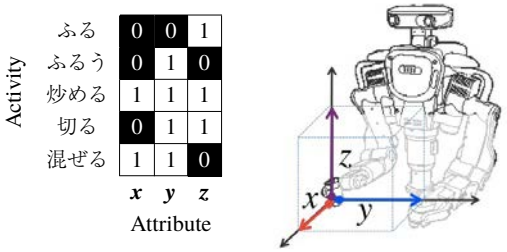



図４　HIRONXC を用いた AAM 作成

表 1 に各関節が動く範囲を示す．ほぼ全ての関節で，1 秒で任意の角度に移動可能なことがわかる．

Jython を使い，HIRONXC を連続で動作させる場合，1 動作につき以下のフォーマットでの入力が必須である．

```
[[[CY, NY, NP],
 [RSY, RSP, REP, RWY, RWP, RWR],
 [LSY, LSP, LEP, LWY, LWP, LWR],
 [RH1, RH2, RH3,RH4], [LH1, LH2,LH3,LH4]], T]
```

つまり，全ての関節角情報を HIRONXC に対して与えなければならない．

#### 3.3.1　動作構成

関節情報は可読性が低く，また人手でロボット動作を作成する際にも不便である．そこで，ロボットを基本動作からより複雑な動作を構成するために，Cheng ら [28] による Activity-Attribute Matrix を参考にする．

#### 3.3.2　Activity - Attribute Matrix

Activity-Attribute Matrix（以下 AAM）は動作と動作に関連している意味属性を符号化したものである．具体例を図 4 で示す．

$M$ を Activity（活動），$N$ を Attribute（属性）とし，各要素 $a_{ij}(i \in M, j \in N)$ において Attribute の Activity への含有関係について Activity $i$ を構成するのに Attribute $j$ が用いられている場合は 1，用いられていない場合 Attribute $j$ は 0 とする，

表 2　時系列対応 AAM の概要

| | $p_1$ | | | | $p_2$ | | | | … | $p_n$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 速く切る | 0 | 0 | -5 | 8 | 0 | 3.5 | 5 | 8 | | 0 | 3.5 | 5 | 8 |
| 速く混ぜる | 0 | 3.5 | 0 | 8 | 3.5 | -3.5 | 0 | 8 | | -3.5 | 0 | 0 | 8 |
| ゆっくり切る | 0 | 0 | -5 | 3 | 0 | 3.5 | 5 | 3 | | 0 | 3.5 | 5 | 3 |
| 細かく切る | 0 | 0 | -5 | 5 | 0 | 1 | 5 | 5 | | 0 | 1 | 5 | 5 |
| ザクザク混ぜる | 0 | 6 | 0 | 8 | 6 | -6 | 0 | 8 | | -6 | 0 | 0 | 8 |
| | $p_{x_1}$ | $p_{y_1}$ | $p_{z_1}$ | $p_{s_1}$ | $p_{x_2}$ | $p_{y_2}$ | $p_{z_2}$ | $p_{s_2}$ | … | $p_{x_n}$ | $p_{y_n}$ | $p_{z_n}$ | $p_{s_n}$ |
| | ↑ | ↑ | ↑ | | ↑ | ↑ | ↑ | | | ↑ | ↑ | ↑ | |
| | $x$ | $y$ | $z$ | | $x$ | $y$ | $z$ | | | $x$ | $y$ | $z$ | |
| CY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| NY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| NP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RSY | 0.1 | 1.8 | 0.7 | | 0.1 | 1.8 | 0.7 | | | 0.1 | 1.8 | 0.7 | |
| RSP | -2.3 | 0.7 | 0.1 | | -2.3 | 0.7 | 0.1 | | | -2.3 | 0.7 | 0.1 | |
| REP | 2.1 | -0.5 | -2.7 | | 2.1 | -0.5 | -2.7 | | | 2.1 | -0.5 | -2.7 | |
| RWY | 0.0 | 0.0 | 0.1 | | 0.0 | 0.0 | 0.1 | | | 0.0 | 0.0 | 0.1 | |
| RWP | 0.2 | 0.2 | 2.7 | | 0.2 | 0.2 | 2.7 | | | 0.2 | 0.2 | 2.7 | |
| RWR | 0.0 | -1.8 | 0.0 | | 0.0 | -1.8 | 0.0 | | | 0.0 | -1.8 | 0.0 | |
| LSY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LSP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | … | 0.0 | 0.0 | 0.0 | |
| LEP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWR | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH1 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH2 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH3 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH4 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH1 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH2 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH3 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH4 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |

$M \times N$ 行列により動作を表現する．図 4 では，Activity として調理動作を例に捉え，それに伴い Attribute として図 4 の右図ように右手の指先を前後に動かす ($\boldsymbol{x}$)，左右に動かす ($\boldsymbol{y}$)，上下に動かす ($\boldsymbol{z}$) を基本ベクトルを設定した．例えば，ふる動作は，$\boldsymbol{x} = 0, \boldsymbol{y} = 0, \boldsymbol{z} = 1$ となっているので，上下に動かす動きであることが分かり，また，切る動きは $\boldsymbol{x} = 0, \boldsymbol{y} = 1, \boldsymbol{z} = 1$ より，左右と上下の動きが含まれていることが分かる．

### 3.3.3 時系列対応 AAM

ロボットを実際に動かすにはそれぞれの関節角を指定する必要がある．そこで本論文では，AAM を改良した「時系列対応 AAM」を提案する．時系列対応 AAM は，Attribute それぞれに対して，係数となるベクトルとの内積を取ることにより，ロボットの動作生成を可能にし，さらに Activity を生成する過程において，それぞれの Attribute の度合い，時系列性，速度等が重要となるため，単位時間あたりの各 Attribute の変動割合にその動作を行うスピードを組み合わせる．本研究では，右手の指先の動き $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ を Attribute とし，単位時間あたりの $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ の変動割合に，その動作を行うスピード $p_s = [0, 10]$ を組み合わせた $[p_x, p_y, p_z, p_s]$ を時系列に $n$ 個並べた $[p_1, p_2, \cdots, p_n] = [[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}], [p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}], \cdots, [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]]$ をロボットに与えることにより，動作の生成を可能にする．表 2 に本研究の時系列対応 AAM を示す．次章にて表 2 の使用法を具体例と共に示す．

### 3.3.4 動作生成

ロボットの動作生成の手順を以下に示す．

1. **基本行列の準備**
   各要素動作の基本ベクトルを並べた基本行列を準備する
2. **係数ベクトルの準備**
   1. で準備した各基本ベクトル方向にどの程度移動させるかを示すベクトル（係数ベクトル）を生成する
3. **基本行列と係数ベクトルの内積**
   1. の基本行列と 2. で生成された係数ベクトルの内積を計算し，関節角座標（以下，「座標」と呼ぶ）を求める
4. **移動先座標の取得および動作時間の追加**
   3. で計算された座標に現在の座標を加えて，移動先の座標を取得する．スピードから時間 $T$ を計算し，ロボットに与える座標のフォーマットは以下の様になる．

```
[[[CY, NY, NP],
 [RSY, RSP, REP, RWY, RWP, RWR],
 [LSY, LSP, LEP, LWY, LWP, LWR],
 [RH1, RH2, RH3,RH4], [LH1, LH2,LH3,LH4]], T]
```

なお時間 $T$ は，表 1 の関節の動作最大速度より，多くの関節でどの角度にも 1 秒以内で動作可能なため，$t_n = 10 - p_{s_n}$ とする．具体例を上記の順に従って以下に示す．

1. **基本行列の準備**
   要素動作 $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ の各基本ベクトルを並べた基本行列の準備のため，ロボットを実際に $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ の各方向に 1 cm ずつ動かし，その時の各関節角の変化を測定する．その結果，準備される基本行列は，表 2 の「↑」下の $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ に紐づく値となる．
2. **係数ベクトルと動作速度の組の生成**
   係数行列の生成を調理動作「切る」を例に説明する．
   切る動作は常に $\boldsymbol{x}$ が 0 で（前後の動きはなく），まず $\boldsymbol{z}$ が負（下へ）の動きをし，次に $\boldsymbol{z}$ が正（上へ）の動きをしながら $\boldsymbol{y}$ が正（左へ）動く，という特徴から生成される．このとき，動作の違いは特に $\boldsymbol{y}$（左右の動き）と速度 $p_s$ によって種類を生成できる．表 2 を見ると，複数の種類の「切る」という動作が，$\boldsymbol{y}$ と $p_s$ の値によって差別化されていることがわかる．
   次に，「速く切る」という動作を見てみると，
   $[[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}], [p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}], \cdots, [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]]$
   の値は [[0,0,-5,8],[0,3.5,5,8],⋯,[0,3.5,5,8]] となっており，時間の経過と共にその動作を示すベクトルが表示されていることがわかる．
3. **基本行列と係数ベクトルの内積，移動先座標の取得および動作時間の追加**
   表 2 中，「速く切る」における最初の係数ベクトルと動作速度の組は，$[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}] = [0, 0, -5, 8]$ である．そのうちの係数ベクトル $[p_{x_1}, p_{y_1}, p_{z_1}]$ と基本行列の内積の結果に現在の座標，つまり初期状態の

```
Initialpose = [[0.0, 0.0, 0.0],
[-0.6,  0, -100,  15.2,  9.4,  3.2],
[ 0.6,  0, -100, -15.2,  9.4, -3.2],
[0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]]
```

を加え，さらに動作のスピード $p_{s_1}$ から計算した動作時間 $t_1 = 10 - p_{s_1}$ を組み合わせ，以下のようになる．

```
 [[[0.0, 0.0, 0.0],
[-4.1, -0.4, -86.4, 14.7, -3.9, 3.2],
```

初期状態　$p_1$ 終了時　$p_2$ 終了時

図５「速く切る」の動作例

```
[0.6, 0.1, -100, -15.2, 9.4, -3.2],
[0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]], 2.0]
```

この動作生成を引き続き $[p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}] = [0, 3.5, 5, 8]$ にも適用した結果のロボットの動作を図 5 に示す．

初期状態と $p_1$ 終了時を比べると，ロボットの右手が下に降りており，次の $p_1$ 終了時と $p_2$ 終了時を比べると，ロボットの右手が左斜め上にあがっており，ロボットで切る動作が生成できていることが確認出来る．

## 4. ロボット動作制御モデルの構築

本章では，まず本研究におけるニューラルネットワークを用いた学習のための適切な誤差最適化手法について検証する．次に，曖昧表現と動作表現の組み合わせによって，生成される動作が変化するモデルを 4 種類構築し，そのモデルの比較を行う．本構築モデルは，入力により特定の出力が変化するモデルである．自然言語処理分野では，Collobert ら [29] が様々な問題に適用可能な共通の言語モデルの構築を深層学習を用いた Multitask Learning によって目指した．また，Kingma ら [30] は一部ラベル付けされた MNIST や SVHN のデータを用い，VAE（Variational AutoEncoder）[31] によって自動に入力の特徴を捉え，文字を特徴に合わせて出力するモデルを構築した．本研究ではこれら先行研究を参考にモデルを構築する．また，入力をラベルとする場合と word2vec による分散表現とする場合についての比較を行い，分散表現の有用性についての検証も行う．なお，本研究では入力の検証とモデルの検証であるため，出力である動作は単純な動作とした．

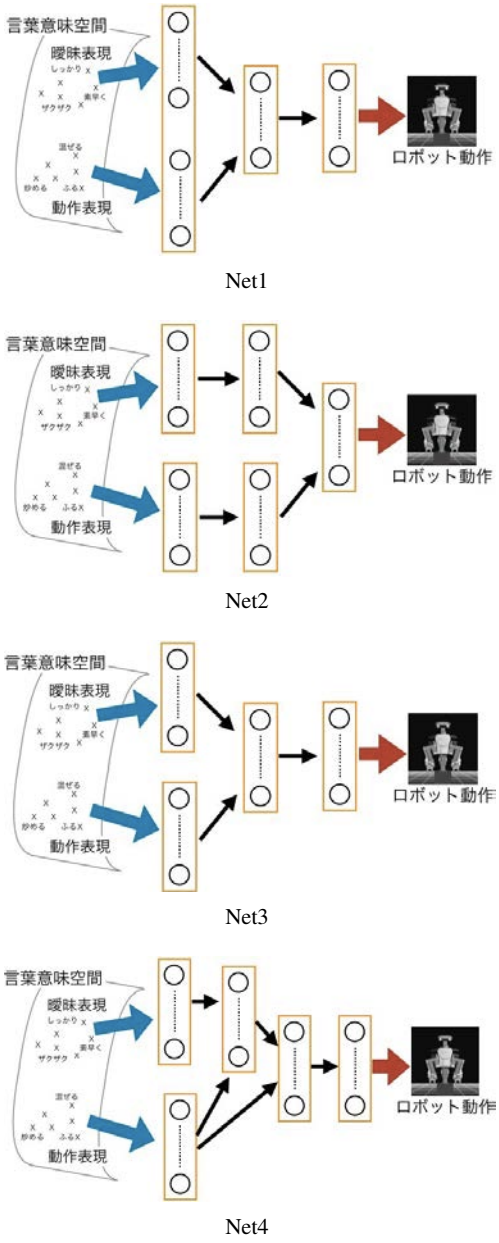



図６　Network の概要

### 4.1 実験仕様

言葉と動作の関係を学習するために，ニューラルネットワーク（NN）を用いる．まず，最適化手法の検証として，図 6 に示したネットワークのうち Net1 について，すべての訓練データに対する誤差関数 $\boldsymbol{E}(w) = \sum_{n=1}^{N} E_n(w)$ を最小化するバッチ学習と，データの一部を使って重み $E_n(w)$ の更新を行う確率的勾配降下法を検証する．またネットワークの検証は Net1 から Net4 の 4 種類を用い，入力をラベル，つまり One-hot のベクトルとした場合と word2vec で作った分散意味表現とした場合とで比較した．なお，Net1 の中間層での活性関数はシグモイド関数，Net2，Net3，Net4 の中間層での活性化関数はソフトプラス関数とする．

**Net1**：曖昧表現と動作表現を結合して一つのベクトルとして入力し，ロボット動作を出す 3 層のニューラルネットワーク．中間層での活性化関数はシグモイド関数を用いる．

**Net2**：曖昧表現と動作表現からそれぞれの中間層を経た後足し合わせ，ロボット動作を出すネットワーク．

**Net3**：曖昧表現と動作表現のそれぞれの中間層を足し合わせ，中間層を経た後ロボット動作を出すネットワーク．

**Net4**：動作表現によって曖昧表現が変化すると仮定する．まず，曖昧表現と動作表現のそれぞれの中間層を足し合わせ，出力された動作により変化した曖昧表現の中間層と，動作表現の中間層を足し合わせ，中間層を経た後ロボット動作を出すネットワーク．

ネットワークの学習方法は誤差逆伝播法を用い，またロボット動作の出力では活性化関数にシグモイド関数を用いた．言葉は，クックパッドのレシピに出現する調理動作表現で最も多い「切る」と 2 番目に多い「混ぜる」に係り受けしている曖昧表現のうち，動作の大きさや速さを表す曖昧表現 12 個（「さっ

表 3　動作の時系列対応 AAM

| | $p_1$ | | | | $p_2$ | | | | $p_3$ | | | | $p_4$ | | | | $p_5$ | | | | $p_6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 動作 | $p_{x_1}$ | $p_{y_1}$ | $p_{z_1}$ | $p_{s_1}$ | $p_{x_2}$ | $p_{y_2}$ | $p_{z_2}$ | $p_{s_2}$ | $p_{x_3}$ | $p_{y_3}$ | $p_{z_3}$ | $p_{s_3}$ | $p_{x_4}$ | $p_{y_4}$ | $p_{z_4}$ | $p_{s_4}$ | $p_{x_5}$ | $p_{y_5}$ | $p_{z_5}$ | $p_{s_5}$ | $p_{x_6}$ | $p_{y_6}$ | $p_{z_6}$ | $p_{s_6}$ |
| ふる | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 |
| ふるう | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 |
| 炒める | 3.5 | 0 | 0 | 5 | -3.5 | 0 | 3 | 5 | 0 | 0 | -3 | 5 | 3.5 | 0 | 0 | 5 | -3.5 | 0 | 3 | 5 | 0 | 0 | -3 | 5 |
| 切る | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 |
| 混ぜる | 0 | 3.5 | 0 | 5 | 3.5 | -3.5 | 0 | 5 | -3.5 | 0 | 0 | 5 | 0 | 3.5 | 0 | 5 | 3.5 | -3.5 | 0 | 5 | -3.5 | 0 | 0 | 5 |
| つぶす | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 |

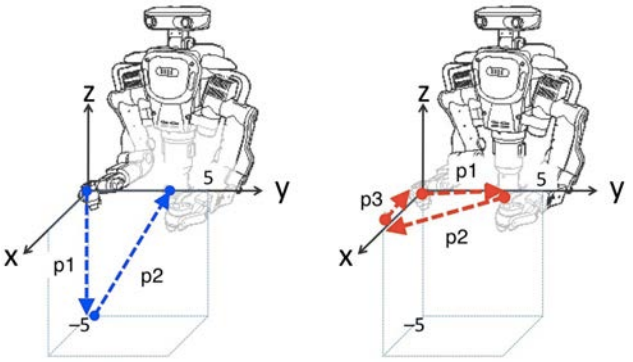



図 7　ロボット動作（左：切る，右：混ぜる）

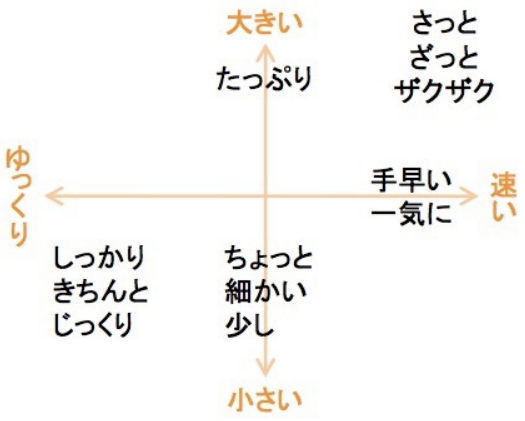



図 8　曖昧表現の動作の程度変化

と」「ざっと」「ザクザク」「たっぷり」「手早い」「一気に」「少し」「細かい」「ちょっと」「しっかり」「きちんと」「じっくり」）と，その曖昧表現と組み合わせて使える調理に関する動作表現 6 個（「ふる」「ふるう」「炒める」「切る」「混ぜる」「つぶす」）を対象とし，ラベルの場合は，曖昧表現は 12 次元，動作表現は 6 次元で表し，分散表現の場合は word2vec の skip-gram を用いて，1 単語を 100 次元の分散表現で表した．動作は，3.3.1 節に示した時系列対応 AAM を用いて人手で作成し，動作表現 6 個について表 3 に示すような $p_n = [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]$ の 4 次元からなる動作を用い，$p_1$ から $p_6$ の連続する 6 つの動作により，24 次元のベクトルで表した．

表 3 で示した動作のうち「切る」と「混ぜる」は図 7 のように表現出来る．

「切る」の動きを例にとると，$p_1$ の動きは，$p_1$=$[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}]$ = $[0, 0, -5, 5]$，次の $p_2$ の動きは，$p_2$=$[p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}]$ = $[0, 3.5, 5, 5]$ で表されており，確かに図 7 左でも $p_1$ で $\boldsymbol{z}$ へ，$p_2$ で $\boldsymbol{y}$ と $\boldsymbol{z}$ への動きを同時に行っている．曖昧表現による動作の程度変化は，例えば「さっと」という動きは「速」くて「大きい」動作の程度であると仮定し，表 3 の着色部や速さを表す $p_s$ を大きくすることで表現する．他の曖昧表現も図 8 のように程度変化を仮定し，それぞれ表 3 の着色部や $p_s$ を変更することで動作を変化させる．なお，動作によって変化する $p_x, p_y, p_z$ は異なる．また，$p_x, p_y, p_z$ の大きさは [-10, 10] の範囲に限定し作成する．

以上を用いて，ネットワークを構築する．入力層のノード数は Net1 ではラベルの場合は 18，word2vec の場合は 200，Net2，Net3，Net4 ではラベルの場合は 12 と 6，word2vec の場合は 100 と 100，出力層のノード数を $p_1 = [p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}]$ から $p_6 = [p_{x_6}, p_{y_6}, p_{z_6}, p_{s_6}]$ の 24 とし，中間層のノード数は 2 から 30 を試行し，最も平均二乗誤差が小さくなった 20 とする．訓練データとして，6 個の動作表現と，12 個の曖昧表現のうちいずれか 7 個をそれぞれ組み合わせた全 42 種の言葉に対し，各 100 個の動作を仮定した動作程度変化になるように切断正規分布により生成し，全 4200 個の動作を作成した．学習をした後，訓練データとして与えた曖昧表現と動作表現の組み合わせの 42 種，未知の言葉として訓練データにない動作表現と曖昧表現の組み合わせの 30 種の動作を評価した．

## 4.2　実験結果

表 4 から表 8 は式 1 で示す誤差率を算出し，パーセントで換算した結果である．予測動作と生成結果で出力された動作をそれぞれロボットに動作させ，$p_1$ から $p_6$ の各時間終了時の予測動作と生成結果動作結果の距離の差（単位：cm）を予測動作の移動距離で割った割合と，その動作にかかる時間の予測動作と生成動作結果の誤差（単位：秒）を予想動作の時間で割った割合を足し合わせてその平均を取る．

$$\frac{1}{12}\sum_{i=1}^{6}\left(\frac{\sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2 + (Z_i - z_i)^2}}{\sqrt{X_i^2 + Y_i^2 + Z_i^2}} + \frac{|T_i - t_i|}{T_i}\right) \quad (1)$$

ここで $X_i$，$Y_i$，$Z_i$，$T_i$ は時刻 $p_i$ で予想した動き，$x_i$，$y_i$，$z_i$，$t_i$ は生成結果の動きである．表 4，表 5，表 7，表 8 は入力を word2vec のベクトルとした場合についてのそれぞれの動作についての誤差率を表しており，着色部は曖昧表現と動作表現の未知の組み合わせを与えた場合である．また表 6 は入力をラベル，つまり One-hot にした場合と word2vec にした場合のそれぞれについて，ネットワーク毎の誤差率の平均を表している．

表４ Net1 バッチ学習

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 65.5 | 59.0 | 56.8 | 56.4 | 57.1 | 55.6 |
| | ざっと | 70.4 | 64.6 | 62.6 | 62.2 | 63.1 | 61.7 |
| | ザクザク | 74.0 | 69.0 | 67.0 | 66.6 | 67.8 | 66.5 |
| | たっぷり | 57.6 | 51.0 | 49.2 | 48.7 | 49.5 | 47.8 |
| | 手早い | 69.2 | 58.1 | 55.9 | 55.5 | 55.1 | 52.3 |
| | 一気に | 75.0 | 63.6 | 61.6 | 61.1 | 60.7 | 57.8 |
| | 少し | 86.5 | 59.9 | 50.0 | 46.7 | 48.3 | 42.3 |
| | 細かい | 74.6 | 52.5 | 47.7 | 46.9 | 45.7 | 38.7 |
| | ちょっと | 68.4 | 50.9 | 47.4 | 47.1 | 45.1 | 40.6 |
| | しっかり | 151.1 | 125.4 | 114.6 | 111.4 | 114.1 | 108.0 |
| | きちんと | 115.6 | 94.1 | 88.9 | 88.0 | 87.0 | 81.0 |
| | じっくり | 93.8 | 76.8 | 73.0 | 72.7 | 71.7 | 66.6 |
| 平均 | | 83.5 | 68.7 | 64.5 | 63.6 | 63.8 | 59.9 |

表５ Net1 確率的勾配降下法

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 32.7 | 42.2 | 45.5 | 27.4 | 40.5 | 36.8 |
| | ざっと | 41.4 | 48.2 | 51.4 | 33.4 | 47.5 | 43.5 |
| | ザクザク | 46.4 | 57.3 | 53.4 | 42.5 | 58.0 | 54.5 |
| | たっぷり | 30.4 | 35.8 | 37.3 | 21.0 | 40.5 | 30.0 |
| | 手早い | 20.4 | 29.8 | 38.3 | 22.0 | 33.1 | 24.6 |
| | 一気に | 29.9 | 35.6 | 45.3 | 25.1 | 41.2 | 26.5 |
| | 少し | 90.2 | 45.1 | 33.7 | 20.2 | 27.5 | 113.3 |
| | 細かい | 60.1 | 26.5 | 23.1 | 16.2 | 24.6 | 48.0 |
| | ちょっと | 31.1 | 17.7 | 28.0 | 18.8 | 25.0 | 32.4 |
| | しっかり | 133.4 | 80.1 | 61.5 | 48.5 | 76.9 | 143.7 |
| | きちんと | 67.1 | 40.8 | 41.7 | 27.4 | 54.8 | 79.3 |
| | じっくり | 53.8 | 30.9 | 36.1 | 23.5 | 50.1 | 41.7 |
| 平均 | | 53.1 | 40.8 | 41.2 | 27.2 | 43.3 | 56.2 |

## 4.3 考察

表 4 と表 5 を比較すると，バッチ学習では全ての動作に対して誤差率が大きくなった．一方，確率勾配降下法を用いた場合には誤差率がバッチ学習に対しては小さくなったが，動作により誤差率のばらつきが見られた．

例えば「切る」動作では，誤差が他の動作に比べて極端に小さくなっている．「切る」動作は動作の程度変化が $p_1$ から $p_6$ の 24 次元のうち，$p_{y_2}$，$p_{s_2}$，$p_{y_4}$，$p_{s_4}$，$p_{y_6}$，$p_{s_6}$ の 6 次元でしか起きておらず，他の動作，例えば「つぶす」動作では $p_{z_1}$，$p_{s_1}$，$p_{z_2}$，$p_{s_2}$，$p_{z_3}$，$p_{s_3}$，$p_{z_3}$，$p_{s_3}$，$p_{z_4}$，$p_{s_4}$，$p_{z_5}$，$p_{s_5}$，$p_{z_6}$，$p_{s_6}$ の 12 次元に比べて少ないためと考えられる．

バッチ学習では，全ての動作がほぼ同じ動きになった．全ての動作の平均的な動きになったため，誤差率のばらつきが少なくなったと考えられる．なお，いずれの場合も同一動作表現と似た曖昧表現の組み合わせた動作についての白色部と着色部の誤差率には大きな違いは見られなかった．言葉の意味空間における曖昧表現間の関係性が，動作に反映されて生成出来ていると考えられ，これは曖昧表現と動作表現をそれぞれベクトルとして入力とすることで可能になっていると想定される．

「さっと」「ふるう」を入力とし，実際にロボットに動作させた結果が図 9 である．バッチ学習より確率的勾配降下法を用いた場合の方が，より動作表現に近い動きができていることが確認出来る．しかし，動作の程度は予想動作と異なっており，曖昧表現の学習が適切に行われているとは言いがたい．

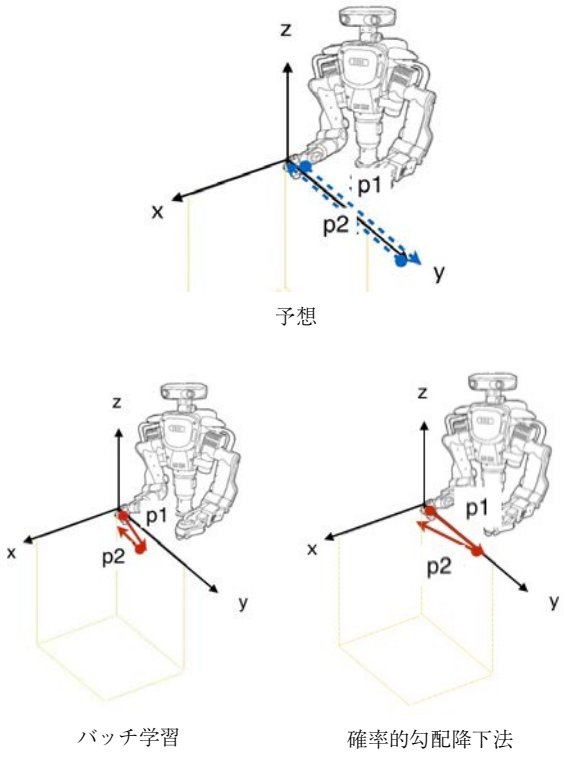



図９ 「さっと」「ふるう」のロボット動作

表６ ネットワーク比較

| Network | 入力 | 誤差率 既知 | 誤差率 未知 |
|---|---|---|---|
| Net1 | word2vec | 40.74 | 47.70 |
| | ラベル | 42.05 | 47.49 |
| Net2 | word2vec | 32.08 | 46.91 |
| | ラベル | 32.13 | 47.78 |
| Net3 | word2vec | 8.44 | 12.71 |
| | ラベル | 8.79 | 14.00 |
| Net4 | word2vec | 8.48 | 12.10 |
| | ラベル | 8.93 | 12.54 |

表 6 より，既知の入力の場合でも未知の入力でも，Net1 の未知の入力を除いて word2vec を用いた場合の方がラベルを用いた場合よりも誤差率が小さくなった．word2vec で生成した言葉の意味空間は本研究で有効であると言える．

ネットワーク構成について，Net1 と比較し，Net2，Net3，Net4 では誤差率が小さくなった．よって，入力の方法として曖昧表現と動作表現を連結した場合より，分けて入力とした場合が有用である．また Net3，Net4 が Net2 よりも誤差率の平均は小さくなっており，動作表現と曖昧表現の言葉を足し合わせた後，ロボット動作を出力する前に一度中間層を経たネットワーク構成の学習が有用であると言える．また Net4 はの未知の入力に対する誤差率の平均が最小であり，未知の入力に対し

表 7　Net2 評価結果

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 30.1 | 30.6 | 26.9 | 18.4 | 29.4 | 31.4 |
| | ざっと | 31.8 | 31.3 | 28.0 | 19.7 | 30.7 | 34.9 |
| | ザクザク | 28.2 | 36.7 | 26.4 | 20.3 | 36.6 | 41.5 |
| | たっぷり | 21.7 | 23.1 | 18.4 | 10.2 | 24.9 | 25.8 |
| | 手早い | 9.7 | 12.7 | 12.4 | 9.1 | 13.8 | 15.9 |
| | 一気に | 8.6 | 10.7 | 13.2 | 8.7 | 12.3 | 11.4 |
| | 少し | 99.3 | 140.2 | 65.6 | 16.1 | 83.6 | 145.7 |
| | 細かい | 79.1 | 52.8 | 25.3 | 13.7 | 48.3 | 43.1 |
| | ちょっと | 39.0 | 31.6 | 14.9 | 9.1 | 27.2 | 26.2 |
| | しっかり | 158.0 | 130.3 | 50.6 | 17.7 | 92.4 | 84.2 |
| | きちんと | 54.4 | 61.4 | 38.8 | 18.3 | 58.8 | 73.5 |
| | じっくり | 46.9 | 29.3 | 15.8 | 13.0 | 31.4 | 23.8 |
| 平均 | | 50.6 | 49.2 | 28.0 | 14.5 | 40.8 | 46.5 |

表 8　Net4 評価結果

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 9.0 | 9.0 | 9.4 | 8.8 | 10.1 | 10.1 |
| | ざっと | 11.3 | 7.9 | 10.6 | 7.5 | 9.0 | 12.7 |
| | ザクザク | 6.1 | 7.5 | 8.9 | 6.5 | 10.4 | 11.2 |
| | たっぷり | 0.9 | 1.2 | 1.7 | 0.8 | 1.6 | 2.5 |
| | 手早い | 8.4 | 10.0 | 9.0 | 9.8 | 10.0 | 10.6 |
| | 一気に | 6.8 | 8.5 | 7.4 | 7.5 | 9.7 | 7.5 |
| | 少し | 18.8 | 37.0 | 9.3 | 1.7 | 13.7 | 28.8 |
| | 細かい | 15.1 | 14.7 | 6.1 | 1.2 | 12.0 | 7.1 |
| | ちょっと | 6.7 | 7.9 | 4.6 | 1.7 | 2.9 | 4.4 |
| | しっかり | 25.9 | 23.1 | 12.3 | 4.8 | 19.9 | 18.2 |
| | きちんと | 14.6 | 17.2 | 14.8 | 6.5 | 22.4 | 15.8 |
| | じっくり | 7.1 | 10.2 | 8.2 | 2.9 | 11.9 | 9.7 |
| 平均 | | 10.9 | 12.9 | 8.5 | 5.0 | 11.1 | 11.6 |

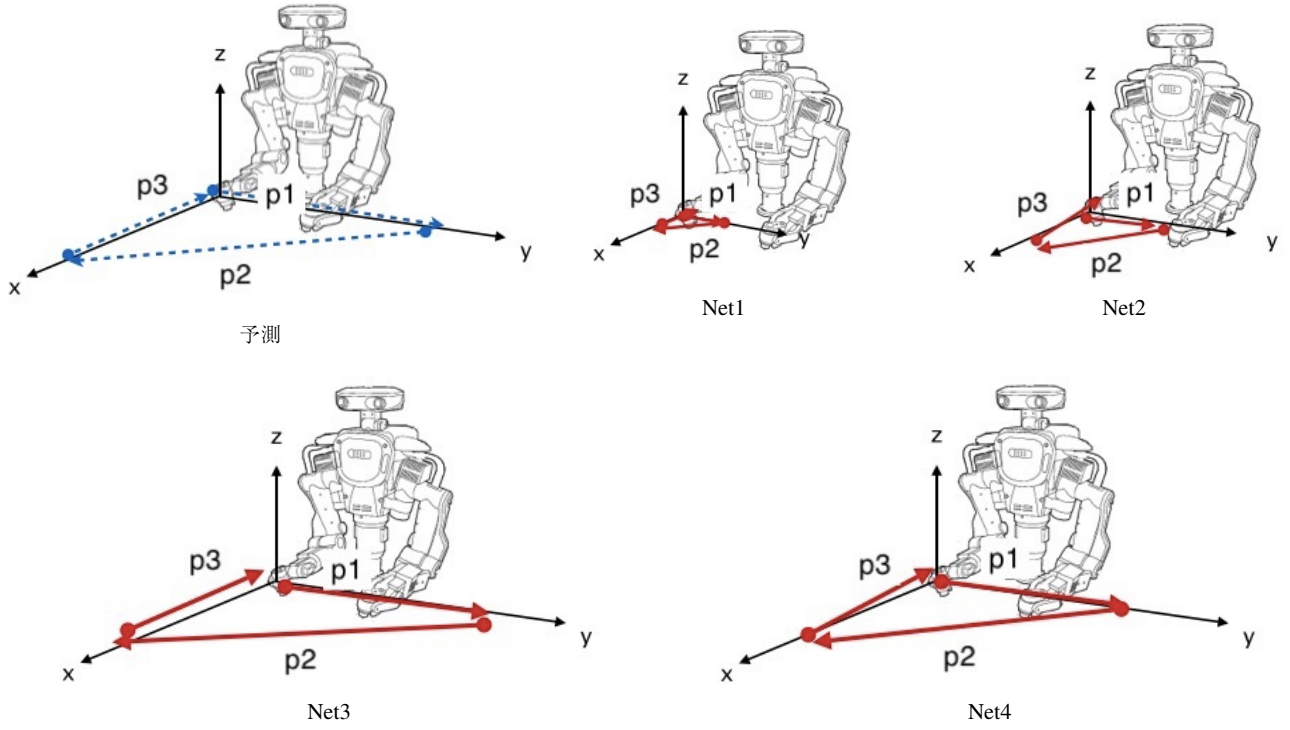



図 10　「ザクザク」「混ぜる」のロボット動作

ては，動作表現によって曖昧表現が変化するという仮定を用いると有効であると言える．

表 7 と表 8 を比較すると，表 7 は，動作表現毎の誤差率の平均が 14.5% から 50.6% となっており，ばらつきがある一方，表 8 では動作表現毎の誤差率の平均が 5.0% から 12.9% と大きくは変わらないことから，Net4 では各動作の特徴を捉えることができていると言える．

また，1 つの曖昧表現における動作表現毎の誤差率について，表 5 や表 7 は，白色部と着色部にばらつきがある一方，表 8 は白色部と着色部に大きな差が見られないことより，曖昧表現が動作毎に変化する学習できていると言える．

「ザクザク」「混ぜる」の全てのネットワークの $p_1$ から $p_3$ までの動作は図 10 のようになる．Net1 に比べ Net2，Net3，Net4 の 3 つの手法は，より「混ぜる」動きができており，特に Net3，Net4 は予想データとより似た動きになることが確認出来る．

また，曖昧表現の入力として全くの未知語として「きちんと」や「しっかり」と word2vec のベクトルがコサイン類似度（それぞれ 0.67，0.66）により意味が近いと判断された「ちゃんと」を「切る」と一緒に与えた場合と「きちんと」「切る」を入力とした場合，また，「ザクザク」「切る」と近い表現として，「キャベツ」を「切る」という入力を与えた場合と「ザクザク」「切る」を入力とした場合のそれぞれの動作結果は図 11 である．「ちゃんと」を入力とした場合も「キャベツ」を入力とした場合も，それぞれの近い言葉と似た動きであるが，わずかな差異

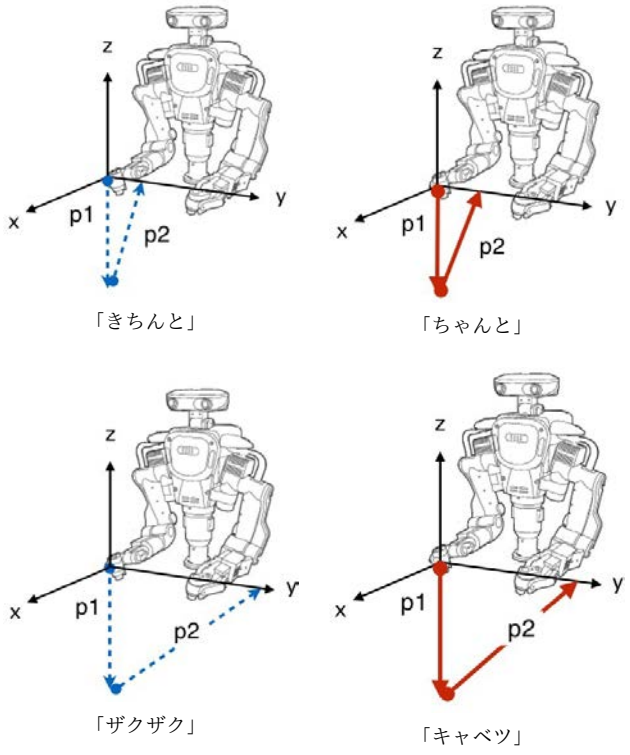



図 11　未知語入力時の「切る」ロボット動作

をもつ動作が生成がされた．言葉意味空間から曖昧表現による動作の変化を推測し，動作生成できていると言える．

## 5. おわりに

本研究では，複数の動作生成に適したネットワークについて検証した．学習する際の最適手法は，確率的勾配降下法がバッチ学習に比べ大きく効果があった．一部の言葉と動作の関係が既知の場合，ラベルを入力にした場合も word2vec を入力とした場合も，曖昧表現と動作表現の組み合わせが未知の場合に動作推定可能であり，特に word2vec を用いた場合は言葉をラベルとして与えた場合に比べ誤差が小さくなり，言葉の分散意味表現の有用性を確認した．また，言葉と動作の関係学習に用いたネットワークの構成による誤差の違いを結果として得，入力層について，曖昧表現と動作表現を連結した場合に比べそれぞれ入力を行った方が良い結果となった．また，適切な中間層の形を得，さらに動作表現により曖昧表現が変化するという仮説が示された．なお入力に word2vec を用いたネットワークについて，全く行ったことのない曖昧表現の動作を生成することができ，既知の表現からの推測ができていると考えられる．

今後の課題として，肘を上げたまま動かす，動きを連続させるなどのより複雑なロボット動作への適用，「にんじんを小さく切る」，「ごぼうを小さく切る」で大きさが変わるように，動作対象に対して動作が変わると考えられるため 3 語以上の入力への対応，「きれいに」や「ちゃんと」のような，より曖昧な言葉の入力に対しての動作生成が挙げられる．また，家庭内で使う場合には口頭での指示が考えられ，非文法文や音声認識誤りや，指示を行う人によって期待する行動が異なる場合が想定される．その対応として，過去の動作からの推定や，周りの状況を判断できるような仕組みの導入が必要と考えられる．なお本研究では word2vec を用いているため，レシピコーパスに出現していない単語に対して，ゼロ頻度問題が起こりうるが，例えば subword の組み合わせ [32, 33, 34] によって，対応できると考えられる．

### 謝辞

本研究は科学研究費補助金（26280096）の支援を受けた．また，クックパッドのコーパスを利用させて頂きました．関係各位に感謝申し上げます．

### 参 考 文 献

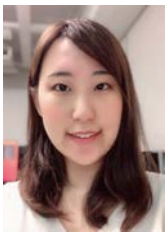
[1] Y. Yamashita and J. Tani: “Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment,” *PLoS Comput Biol*, Vol.4, No.11, p. e1000220, 2008.
[2] Y. Sugita and J. Tani: “Learning semantic combinatoriality from the interaction between linguistic and behavioral processes,” *Adaptive Behavior*, Vol.13, No.1, pp. 33-52, 2005.
[3] T. Ogata, M. Murase, J. Tani, K. Komatani, and H. G. Okuno: “Two-way translation of compound sentences and armmotions by recurrent neural networks,” *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2007)*, pp. 1858-1863. IEEE, 2007.
[4] F. Stramandinoli, D. Marocco, and A. Cangelosi: “The grounding of higher order concepts in action and language: a cognitive robotics model,” *Neural Networks*, Vol.32, pp. 165-173, 2012.
[5] D. Bahdanau, K. Cho, and Y. Bengio: *Neural Machine Translation by Jointly Learning to Align and Translate*, 2014.
[6] I. Sutskever, O. V. inyals, and Q. V Le: “Sequence to sequence learning with neural networks,” *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pp. 3104-3112, 2014.
[7] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao: “Short text clustering via convolutional neural networks,” *Proc. of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 62-69, 2015.
[8] J. Hammerton: “Named entity recognition with long short-term memory,” *Proc. of the Seventh Conf. on Natural Language Learning at HLT-NAACL 2003*, pp. 172-175, 2003.
[9] E. Stroh and P. Mathur: *Question Answering Using Deep Learning*, 2016.
[10] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean: *Google’s Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation*, 2016.
[11] V. Mnih, N. Heess, A. Graves, and K. kavukcuoglu: “Recurrent models of visual attention,” *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pp. 2204-2212, 2014.
[12] K. Cho, B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio: “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724-1734, 2014.
[13] S. Hochreiter and J. Schmidhuber: “Long short-term memory,” *Neural Computation*, Vol.9, No.8, pp. 1735-1780, 1997.
[14] M. Lesk: “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone,” *Proc. of the 5th annual Int. Conf. on Systems documentation*, pp. 24-26, 1986.
[15] C. Leacock, G. A Miller, and M. Chodorow: “Using corpus statistics and wordnet relations for sense identification,” *Computational*

*Linguistics*, Vol.24, No.1, pp. 147-165, 1998.
[16] H. Schütze: "Automatic word sense discrimination," *Computational linguistics*, Vol.24, No.1, pp. 97-123, 1998.
[17] D. Yarowsky: "Hierarchical decision lists for word sense disambiguation," *Computers and the Humanities*, Vol.34, No.1-2, pp. 179-186, 2000.
[18] X. Chen, Z. Liu, and M. Sun: "A unified model for word sense representation and disambiguation," *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1025-1035, 2014.
[19] I. Iacobacci, M. T. Pilehvar, and R. Navigli: "Embeddings for word sense disambiguation: An evaluation study," *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 897-907, 2016.
[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean: *Efficient Estimation of Word Representations in Vector Space*, CoRR, Vol.abs/1301.3781, 2013.
[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean: "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, pp. 3111-3119, 2013.
[22] G. E. Hinton: *Distributed representations*, 1984.
[23] O. Levy and Y. Goldberg: "Neural word embedding as implicit matrix factorization," *Advances in neural information processing systems*, pp. 2177-2185, 2014.
[24] T. Kudo, K. Yamamoto, and Y. Matsumoto: "Applying conditional random fields to japanese morphological analysis," *EMNLP*, Vol.4, pp. 230-237, 2004.
[25] Z. S Harris: "Distributional structure," *Word*, Vol.10, No.2-3, pp. 146-162, 1954.
[26] W. Yih, G. Zweig, and J. Platt: "Polarity inducing latent semantic analysis," *Proc. of the 2012 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1212-1222, 2012.
[27] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith: "Retrofitting word vectors to semantic lexicons," *Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1606-1615, 2015.
[28] H. Cheng, F. Sun, M. L. Griss, P. Davis, J. Li, and D. You: "Nuactiv: Recognizing unseen new activities using semantic attribute-based learning," *Proc. of the 11th annual Int. Conf. on Mobile systems, Applications, and Services*, pp. 361-374, 2013.
[29] R. Collobert and J. Weston: "A unified architecture for natural language processing: Deep neural networks with multitask learning," *Proc. of the 25th Int. Conf. on Machine Learning*, pp. 160-167. ACM, 2008.
[30] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling: "Semi-supervised learning with deep generative models," *Advances in Neural Information Processing Systems*, pp. 3581-3589, 2014.
[31] D. P. Kingma and M. Welling: *Auto-encoding Variational Bayes*, arXiv preprint arXiv:1312.6114, 2013.
[32] J. Zhao, S. Mudgal, and Y. Liang: "Generalizing word embeddings using bag of subwords," *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Proc.*, pp. 601-606, 2018.
[33] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov: "Enriching word vectors with subword information," *Trans. of the Association for Computational Linguistics*, Vol.5, pp. 135-146, 2017.
[34] Y. Pinter, R. Guthrie, and J. Eisenstein: "Mimicking word embeddings using subword RNNs," *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Proc.,* pp. 102-112, 2017.

（2018 年 11 月 15 日　受付）
（2019 年 10 月 4 日　採録）

［問い合わせ先］
〒112-8610　東京都文京区大塚 2-1-1
お茶の水女子大学 理学部 3 号館 506 室 小林研究室
濵園　侑美
TEL: 03-5978-5708
E-mail: hamazono.yumi@is.ocha.ac.jp

## 著 者 紹 介



**濵園（はまぞの）　侑美（ゆみ）**［非会員］

2015 年お茶の水女子大学理学部情報科学科卒業．2017 年同大学院人間文化創成科学研究科理学専攻博士前期課程修了．現在，同大学院人間文化創成科学研究科理学専攻博士後期課程在学中．

**小林（こばやし）　一郎（いちろう）**［正会員］

1995 年東京工業大学大学院総合理工学研究科システム科学専攻博士課程単位取得退学．同年博士（工学）．同年法政大学経済学部研究助手．1996 年から 2003 年まで法政大学経済学部助教授．2000 年から 2005 年まで理化学研究所脳科学総合研究センター客員研究員．2003 年お茶の水女子大学理学部情報科学科助教授．2007 年 ドイツ人工知能研究所客員研究員．2007 年 から 2008 年までスタンフォード大学 CSLI 客員研究員．2011 年お茶の水女子大学大学院理学専攻教授．2017 年 より 産業技術総合研究所人工知能研究センター 招聘研究員．．現在お茶の水女子大学基幹研究院自然科学系教授. 自然言語処理，人工知能，機能言語学などの研究に従事．

**麻生（あそう）　英樹（ひでき）**［非会員］

1981 年東京大学工学部計数工学科卒業．1983 年同大学院工学系研究科情報工学専攻修士課程修了．同年通商産業省工業技術院電子技術総合研究所入所．1993 年から 1994 年にドイツ国立情報処理研究センターに客員研究員として滞在．2015 年から国立研究開発法人産業技術総合研究所人工知能研究センター副研究センター長を務める．脳の情報処理モデルへの興味にもとづき，ニューラルネットワーク，統計的機械学習，などの基礎理論・アルゴリズムと，学習能力を持つ知的情報処理システムへの応用に関する研究開発に従事．

**中村（なかむら）　友昭（ともあき）**［非会員］

2009 年電気通信大学大学院電気通信学研究科博士前期課程修了．2011 年同博士後期課程修了．博士（工学）．2011 年日本学術振興会特別研究員（PD）．2013 年ホンダ・リサーチ・インスティチュート・ジャパンリサーチャー．2014 年より電気通信大学情報理工学研究科助教．2019 年より同研究科准教授．知能ロボットの研究に従事．日本ロボット学会，人工知能学会各会員．

長井 隆行［非会員］

1993 年慶應義塾大学理工学部電気工学科卒業．1995 年同大学院理工学研究科電気工学専攻前期博士課程修了．1997 年同後期博士課程修了．博士（工学）．1998 年電気通信大学助手，2003 年カリフォルニア大学サンディエゴ校客員研究員，2015 年電気通信大学教授を経て，2018 年より大阪大学大学院基礎工学研究科教授．電気通信大学人工知能先端研究センター特任教授，玉川大学脳科学研究所特別研究員，産総研人工知能研究センター客員研究員を兼務．IROS Best Paper Award Finalist, Advanced Robotics Best Paper Award，人工知能学会論文賞など多数受賞．知能ロボティクス，認知発達ロボティクス，ロボット学習に関する研究に従事．AI とロボティクスの融合による，人間のように柔軟で汎用的な知能の実現を目指している．

持橋 大地［非会員］

1998 年東京大学教養学部基礎科学科第二卒業．2005 年奈良先端科学技術大学院大学情報科学研究科博士後期課程終了．博士（理学）．ATR 音声言語コミュニケーション研究所，NTT コミュニケーション科学基礎研究所を経て，2011 年より統計数理研究所 准教授．

**Motion Generation Using Humanoid Robot with Language Understanding**

by

**Yumi HAMAZONO, Ichiro KOBAYASHI, Hideki ASOH, Tomoaki NAKAMURA, Takayuki NAGAI, and Daichi MOCHIHASHI**

**Abstract**:

Currently, the advent of the low birthrate and aging society in Japan has become a problem so it will be increased that the opportunity to resolve the problem by using robots at home. Therefore, it is convinced that robots will have more opportunities for being active at home. These days, we can get robots more inexpensively and it makes the communication between human and robot will be significant progress. When the robot is at home, the condition in which the robot can live together with residents is that the robot mimics residents' experiences telling by words and gestures so that learns how or what to do at home. The objective of this study is to make a robot enable to properly behave based on instructions given by people. We therefore consider a way of associating words with robot's actions so that a robot can behave by understanding the meaning of words. As a concrete example, we focus on various types of cooking actions represented by words with adverbial expressions and use multilayer perceptron to learn relation between adverbial expressions and robot's actions. The meaning of elementary cooking instructions is represented with distributed semantics by means of word2vec. To represent the actions of a robot, we have expanded the framework of Activity-Attribute Matrix (AAM) so as it can deal with the motion of actions. We have employed multilayer perceptron to learn the correspondence between those actions and the meaning of the instructions, and confirmed how much the actions that a robot has never done can be precisely estimated with the meaning of the given unknown instructions with the learned model.

**Keywords**: natural language processing, machine learning, humanoid robot, neural network, multitask learning

Contact Address: **Yumi HAMAZONO**
*Kobayashi Lab., Ochanomizu University*
*room 506, Faculty of Science Bld.3, 2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610, Japan*
TEL: +81-3-5978-5708
E-mail: hamazono.yumi@is.ocha.ac.jp