

# 語彙の分散意味表現とロボット動作との対応関係の学習

Learning the Correspondence between Distributed Semantics of Words and Robot's Action

濱園侑美 \*1  
Yumi Hamazono

小林一郎 \*1  
Ichiro Kobayashi

麻生英樹 \*2  
Hideki Aso

持橋大地 \*3  
Daichi Mochihashi

Muhammad Attamimi\*4

中村友昭 \*4  
Tomoaki Nakamura

長井隆行 \*4  
Tagai Takayuki

\*1お茶の水女子大学  
Ochanomizu University

\*2産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology

\*3統計数理研究所  
The Institute of Statistical Mathematics

\*4電気通信大学  
The University of Electro-Communications

Toward ageing society, it is convinced that robots will have more opportunities for being active at home. As a requirement for the robots being able to live together with the residents, it is necessary for them to acquire knowledge through the words and gestures by people, and to share the same experience with them. Considering these things, in this paper, we aim to make a robot be able to properly behave by the instructions given by people. Based on this, we propose a method to associate words with robot's actions so that can guess actions from words. In concrete, we employed multilayer perceptron to learn the correspondence relation between various types of cooking actions modified with an ambiguous expression and the meaning of the instructions, and confirmed how much the actions that a robot has never done can be precisely estimated by the meaning of the given unknown instructions.

## 1. はじめに

少子高齢化社会の到来により、家庭内でロボットを利用することによって問題解決を図ろうとする場面が増えると考えられる。近年では、ロボットを安価に入手出来るようになっており、人間とロボットとのコミュニケーションが大きく進展する可能性がある。家庭内にロボットが入ってきた場合、ロボットが居住者と共に暮らせる条件として、言葉や身振りにより居住者の経験をロボットに伝え、ロボットはそれを真似し学習することが必要になってくると考える。

このことを踏まえ、本研究では人の言葉による指示からロボットが動作を行なうことを目標に、言葉と動作の対応関係を学習することによって、言葉の意味から初めて行なう動作であっても行動出来るようにすることを目的としている。ロボットの動作として、調理に関する動作を対象とする。先行研究 [3] において「切る」という基本動作を収束する副詞の意味を推定したのに対し、本研究では、調理動作の種類を増やすことにより、より多様なロボット動作と曖昧な表現との対応関係についての考察を行なう。

## 2. 提案手法概要

いくつかの言葉の意味と動作表現の対応関係が既知であるとする。この時、動作と対応関係が分からない未知の言葉が与えられた際に、他の言葉との意味的な関係から対応する動作を推定する手法を提案する。図 1 に提案手法の概要を示す。言葉を意味空間へ配置する方法は word2vec[2] を用い、動作を表現空間へ配置する方法は、先行研究 [3] で提案した Activity-Attribute Matrix(AAM)[1] を動作生成に適した形に改良した

連絡先: 濱園侑美, お茶の水女子大学理学部情報科学科, 東京都文京区大塚 2-1-1 理学部 3 号館 506, g1020528@is.ocha.ac.jp

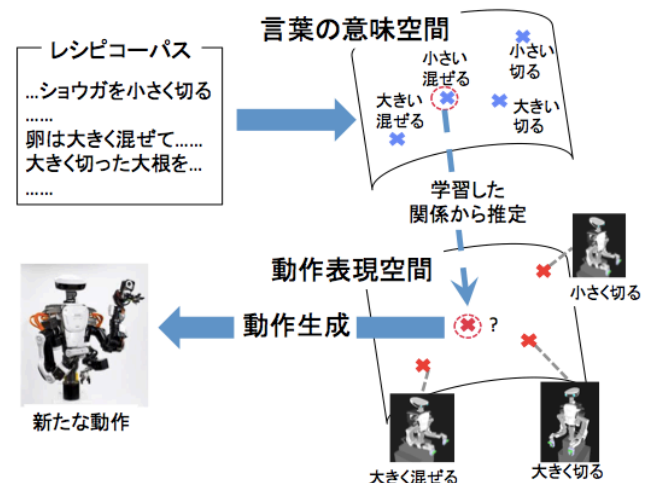


図 1: 提案手法の概要

時系列対応 AAM に対し、時間要素を時間から速度の割合と変更し、それを採用した。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. ロボットの動作

### 3.1 ヒューマノイドロボットの概観

使用するロボットは(株)川田工業社製ヒューマノイドロボット HIRONXC (図 2) \*1 を用いる。HIRONXC は全 24 つの関節を持ち、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。表 1 にそれぞれの関節角が動く範囲を示す。

\*1 <http://robot-support.kawada.jp/support/hiro/>

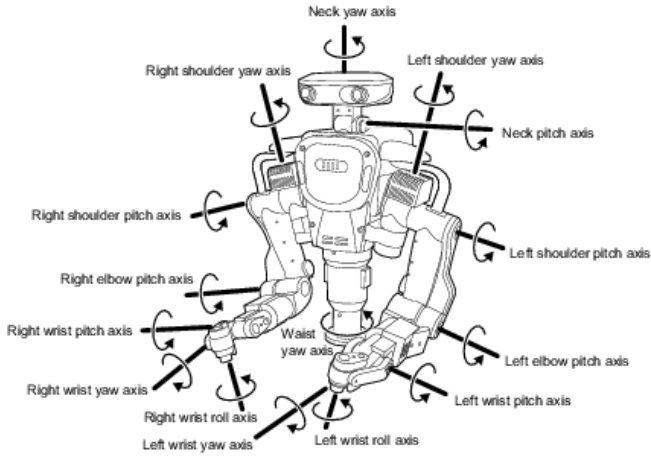


図 2: HIRONXC の関節

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

### 3.2 動作構成

ロボットの調理動作を関節軸の基本動作から構成するために、Cheng ら [1] による Activity-Attribute Matrix(以下 AAM) を参考にする。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 3 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,  $M \times N$  行列により動作を表現する。本研究では, Activity として調理動作を例に捉え, それに伴い Attribute として図 3 の右図のように右手の指先を前後に動かす ( $x$ ), 左右に動かす ( $y$ ), 上下に動かす ( $z$ ) を基本ベクトルを設定した。

また, 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, Attribute の  $x, y, z$  それぞれに対して, 係数となるベクトルとの内積をとることにより, ロボットの動作生成を可能にする。さらに本研究では 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,  $x, y, z$  の変動割合に時間の要素  $t$  を組み合わせ  $[p_x, p_y, p_z, t]$  を時系列に  $n$  個並べた  $[[p_{x_1}, p_{y_1}, p_{z_1}, t_1], [p_{x_2}, p_{y_2}, p_{z_2}, t_2], \dots$

Activity

ふる	0	0	1
ふるう	0	1	0
炒める	1	1	1
切る	0	1	1
混ぜる	1	1	0

Attribute  
 $x \ y \ z$

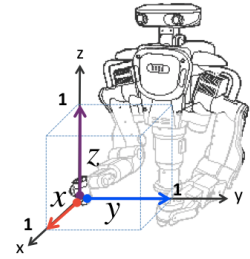


図 3: HIRONXC を用いた AAM 作成

,  $[p_{x_n}, p_{y_n}, p_{z_n}, t_n]$  を与えることにより動作の生成を可能にする。なお, 時間要素  $t$  は本研究では速度とし,  $[0, 1]$  で表した。表 2 に本研究で提案する時系列対応 AAM を示す。

表 2: 時系列対応 AAM の概要

	$s_1$			$t_1$	$s_2$			$t_2$	...	$s_n$			$t_n$
	$p_{x_1}$	$p_{y_1}$	$p_{z_1}$		$p_{x_2}$	$p_{y_2}$	$p_{z_2}$			$p_{x_n}$	$p_{y_n}$	$p_{z_n}$	
速く切る	0	0	-5	0.8	0	3.5	5	0.8	...	0	3.5	5	0.8
速く混ぜる	0	3.5	0	0.8	3.5	-3.5	0	0.8	...	-3.5	0	0	0.8
ゆっくり切る	0	0	-5	0.3	0	3.5	5	0.3	...	0	3.5	5	0.3
細かく切る	0	0	-5	0.5	0	1	5	0.5	...	0	1	5	0.5
ザクザク混ぜる	0	6	0	0.8	6	-6	0	0.8	...	-6	0	0	0.8
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	

ロボットの動作生成の手順を以下に示す。

1. 基本ベクトルの準備: 要素動作となる  $x, y, z$  方向の基本ベクトルを準備する
2. 係数ベクトルの準備: 1. で準備した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. 基本ベクトルと係数ベクトルの内積: 1. の基本ベクトルと 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: 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。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。

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

#### 4. 言葉の分散意味表現

Mikolovら [2] によって提案された word2vec は、単語をベクトルで表現し、同じ文脈の中にある単語はお互いに近い意味を持つように単語をベクトル化して表現する定量化手法である。本研究ではこれを利用し、単語の意味関係から、未知の単語に対する動作の推定を可能にする。コーパスはクックパッド\*2 のレシピを用いた。なお 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 による分かちを正確にし、より良い分散意味ベクトルが構築出来るようにした。曖昧表現の word2vec による言葉の分散意味表現を主成分分析し可視化した結果が図 4 である。

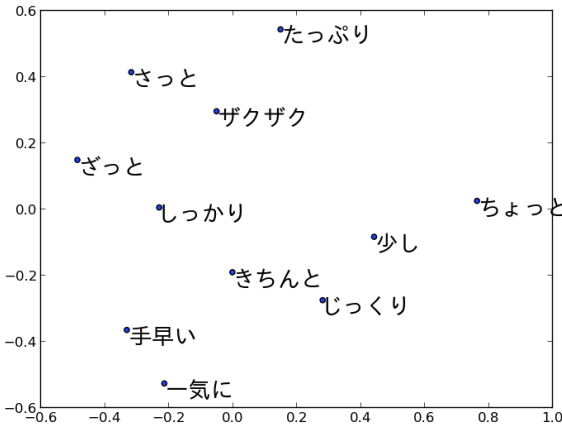


図 4: word2vec による分散意味表現

#### 5. 言葉と動作の対応関係学習

言葉と動作の関係を学習するために、3 層の階層型ニューラルネットワーク (NN) を用い、学習方法としては誤差逆伝播法を用いた。また活性化関数はシグモイド関数を用いた。言葉は word2vec のうち skip-gram を利用し、1 単語を 50 次元の分散意味表現で表した。word2vec は意味がベクトルの演算で表現出来ることが示されている [4] ことから、本研究では動作とその程度を表す曖昧表現のそれぞれ 50 次元のベクトルを結合した 100 次元のベクトルをひとつの意味ベクトルとして扱う。動作は様々なパターンを 3.2 節に示した時系列対応 AAM にて作成し、表 3 に示す、時刻  $s_1, s_2, s_3$  の 3 つの動作を合わせて 12 次元のベクトルで表した。

表 3: 動作の時系列対応 AAM

動作	$s_1$				$s_2$				$s_3$			
	$p_{x_1}$	$p_{y_1}$	$p_{z_1}$	$t_1$	$p_{x_2}$	$p_{y_2}$	$p_{z_2}$	$t_2$	$p_{x_3}$	$p_{y_3}$	$p_{z_3}$	$t_3$
ふる	0	0	-3.5	0.5	0	0	3.5	0.5	0	0	0	0
ふるう	0	3.5	0	0.5	0	-3.5	0	0.5	0	0	0	0
炒める	3.5	0	0	0.5	-3.5	0	3	0.5	0	0	-3	0.5
切る	0	0	-5	0.5	0	3.5	5	0.5	0	0	0	0
混ぜる	0	3.5	0	0.5	3.5	-3.5	0	0.5	-3.5	0	0	0.5
つぶす	0	0	-3.5	0.5	0	0	3.5	0.5	0	0	0	0

表 3 で示した動作のうち「切る」と「混ぜる」は図 5 のように表現出来る。「切る」の動きを例にとると、 $s_1$  の動きは、 $s_1=[p_{x_1}, p_{y_1}, p_{z_1}, t_1] = [0, 0, -5, 0.5]$  で表され、図 5 左の  $s_1$  でも  $z$  への動きを行っている。次の  $s_2$  の動きは、

\*2 <http://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html>

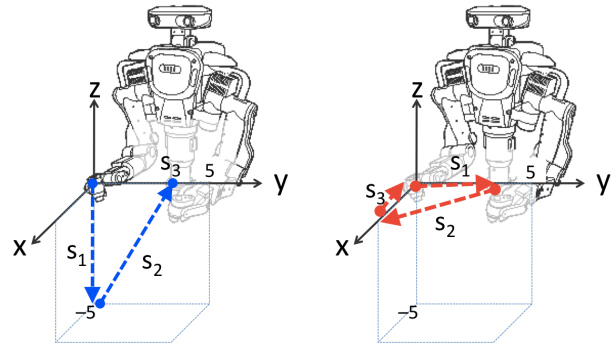


図 5: 時系列対応 AAM を用いた動き (左: 切る, 右: 混ぜる)

$s_2=[p_{x_2}, p_{y_2}, p_{z_2}, t_2] = [0, 5, -3.5, 0.5]$  で表され、これも図 5 左の  $s_2$  で  $y$  と  $z$  への動きを同時に行っている。更に最後の  $s_3$  の動きは、 $s_3=[p_{x_3}, p_{y_3}, p_{z_3}, t_3] = [0, 0, 0, 0]$  で表され、動かないことが図 5 左より確認出来る。また、図 6 のように曖

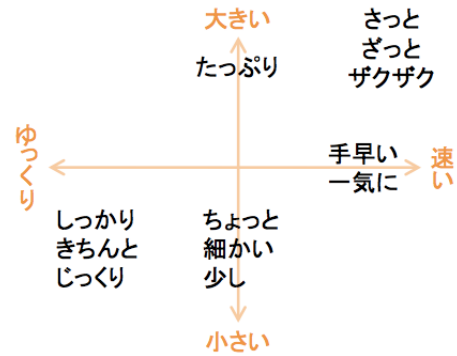


図 6: 想定される曖昧表現の動作程度

昧表現の動作の程度を想定し、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する。表 3 の着色部や速さを表す  $t_n$  を変化させることで、動作の程度を変化させる。

これにより入力層のノード数を曖昧表現と動作表現を合わせた意味ベクトルの 100、出力層のノード数を 12 とし、中間層のノード数を訓練データと構築したモデルの最小二乗誤差が小さいと判断された 49 とする NN を構築した。訓練データとして、6 個の動作表現 (ふる, ふるう, 炒める, 切る, 混ぜる, つぶす) と、12 個の曖昧表現 (さっと, ざっと, ギャクザク, たっぷり, 手早い, 一気に, 少し, ちよっと, しっかりと, きちんと, じっくり) のうちいずれか 7 個の意味ベクトルの 42 個に対し各 100 個の全 4200 個を与えた。なお、訓練データは切断正規分布により作成した。

学習をした後、評価データは訓練データとして与えた 42 個の意味ベクトル、未知の言語表現として訓練データに用いた意味ベクトル以外の 30 個の意味ベクトル (着色部) を入力したところ、表 4 に示す結果になった。

表 4 は予測した動作と、評価結果として出てきた動作をそれぞれロボットに動作させ、 $s_1, s_2, s_3$  での  $x, y, z$  軸に関するそれぞれの誤差 (単位: cm) と、速さの誤差の平均二乗誤差を「誤差」として表している。

表 4 の結果より、訓練データで用いた動作表現と曖昧表現

表 4: 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	13.67	18.87	6.75	1.20	7.70	0.72
	ざっと	13.31	16.39	6.74	1.03	7.47	0.87
	ザクザク	12.68	13.62	5.77	0.84	13.25	1.36
	たっぶり	11.81	9.57	4.99	1.24	4.45	0.44
	手早い	6.74	10.99	4.78	0.62	5.58	1.00
	一気に	6.33	11.31	4.79	0.42	6.06	0.85
	少し	5.01	5.90	4.85	0.05	2.91	2.02
	細かい	2.78	3.79	3.48	0.77	3.86	1.00
	ちょっと	2.52	4.45	3.42	0.13	6.07	1.12
	しっかり	2.87	5.79	3.91	0.25	4.62	1.21
	きちんと	2.78	3.93	3.86	0.02	5.81	0.78
じっくり	2.80	3.58	3.44	0.21	5.20	1.07	

の組み合わせの意味ベクトルと、未知の意味ベクトルとの差は「ザクザク」「混ぜる」を除いては、あまりないと言える。動作表現を比べると、「切る」動作は 12 個の曖昧表現のうち 9 個の表現に対し、誤差が最も小さくなっている。また、「つぶす」動作は「さっと」「ざっと」「たっぶり」の 3 個の曖昧表現に対して、「切る」動作の誤差よりも小さくなっており、他の曖昧表現に関しても、「切る」以外の動作よりも誤差が小さくなっている。一方、「ふるう」動作は、8 つの曖昧表現に対し誤差が最大となった。

最も誤差の大きい「さっと」「ふるう」と、最も誤差の小さい「きちんと」「切る」の予想した動作と実験の結果の動作を、実際にロボットで生成すると図 7, 図 8 のようになった。

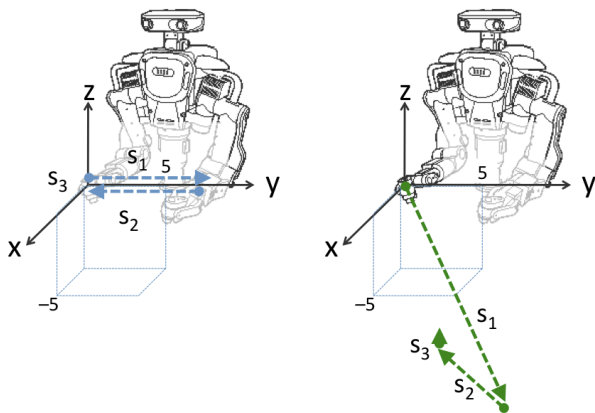


図 7: 「さっと」「ふるう」(左: 予想, 右: 実験結果)

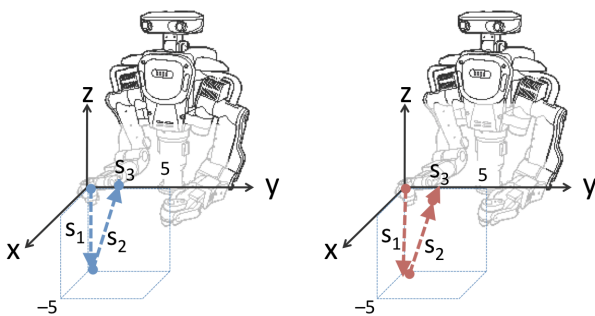


図 8: 「きちんと」「切る」(左: 予想, 右: 実験結果)

図 7 の「さっと」「ふるう」動作において、 $s_1$  の動きで実験結果では  $z$  軸マイナス方向へと動いている。 $s_2$  では、 $z$  軸プラス方向へ動いており、更に  $y$  軸マイナス方向の動きは小さい。 $s_1, s_2$  で  $z$  軸方向の動きが出た理由としては、他の動作で  $z$  軸方向の動きがあるためと考えられる。また、図 8 の「きちんと」「切る」動作において、予想動作と同じような動作が実験結果でも生成出来ている事が分かる。

## 6. まとめ

本研究では、ロボットの動作生成を行うため調理動作生成を例とし、動作生成に対応するため、Cheng らによる Activity-Attribute Matrix を参考にして動作要素の時系列変化を捉える関係行列を作成した。また、言葉と動作の対応関係を取るため、言葉は word2vec を使った分散意味表現をとり、動作表現と曖昧表現の組み合わせである意味ベクトルにより、多様な動作にも対応出来るよう工夫した。作成した分散意味表現と動作のベクトルによるニューラルネットワークを用い、学習を行った。実験の結果より、学習した意味ベクトルから、未知の意味ベクトルの推定が出来ており、言葉の汎用が出来ていると考えられる。動作について、 $z$  軸方向の動きが行なわれている動作では大きな誤差は見られない一方、 $z$  軸方向の動きが行なわれていない動作に関して、大きく誤差が出た。多くの動作で行なわれている  $z$  軸方向の動きが生成時に出てしまうためと考えられる。

今後の課題としては、言葉と動作の対応関係をより明確にするため、動作によって言葉の意味空間を修正し、より多様な言葉から動作の生成を可能にするつもりである。また、動作の学習と曖昧表現の学習を分ける事で、より動作の生成を確かなものにするつもりである。

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] Heng-Tze Cheng, Feng-Tso Sun, Martin Griss, Paul Davis, Jianguo Li, Di You, “NuActiv: Recognizing Unseen New Activities Using Semantic Attribute-Based Learning”, Mobile Systems, Applications, and Services, 2013.
- [2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, International Conference on Learning Representations, 2013.
- [3] 濱園侑美, 小林一郎, 麻生英樹, 持橋 大地, Muhammad Attamimi, 中村友昭, 長井隆行, “語彙の分散意味に基づくヒューマノイドロボットの動作生成に向けた基礎的検討”, 言語処理学会第 22 回年次大会, 2016.
- [4] Omer Levy, Yoav Goldberg, “Neural Word Embedding as Implicit Matrix Factorization”, Neural Information Processing Systems, 2014.