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

理学専攻・情報科学コース　　濱園侑美

1　はじめに

現在，日本は超高齢化社会に突入している．少子高齢化社会の到来による人手不足を，ロボットを利用することで問題解決をはか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発展する可能性がある．家庭内でロボットを用いる場合，ロボットが居住者と協調して暮らすことができる条件として，言葉や身振りを用いいることで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ると考える．

このことを踏まえ，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，初めて行う動作であっても言葉の意味から推測し，動作を行なえるようにすることを目的としている．なお，本研究はロボットの動作として，調理に関する動作を対象とする．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する．

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．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．

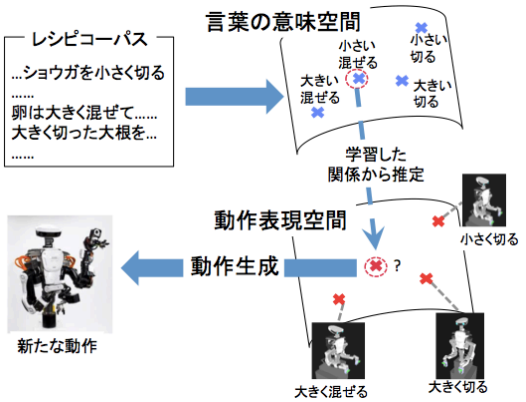

図 1: 提案手法の概要

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，Activity-Attribute Matrix(AAM)[1] を動作生成に適した形に改良した時系列対応 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　ロボットの動作

3.1　ヒューマノイドロボットの概要

（株）川田工業社製ヒューマノイドロボット HIRONXC を用いる．HIRONXC は全 24 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 秒かけて関節を指定された角度へと動かすことが可能である．

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 2 で示す．

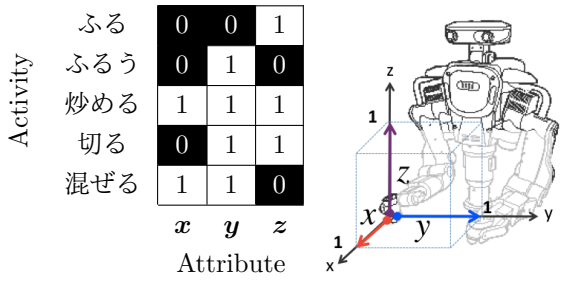

図 2: HIRONXC を用いた AAM 作成

M を Activity，N を Attribute 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 2 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．

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 $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ 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，$\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ の変動割合に時間の要素 t を組み合わせた $[p_x, p_y, p_z, t]$ を時系列に n 個並べた $[[p_{x_1}, p_{y_1}, p_{z_1}, t_1], [p_{x_2}, p_{y_2}, p_{z_2}, t_2], \cdots, [p_{x_n}, p_{y_n}, p_{z_n}, t_n]]$ 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 1 に本研究で提案する時系列対応 AAM を示す．

表 1: 時系列対応 AAM の概要

	s_1				s_2				...	s_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
	p_{x_1}	p_{y_1}	p_{z_1}	t_1	p_{x_2}	p_{y_2}	p_{z_2}	t_2	...	p_{x_n}	p_{y_n}	p_{z_n}	t_n
	↑	↑	↑		↑	↑	↑			↑	↑	↑	
	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$		$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$			$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$	
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	

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[1] のレシピを用いた．

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 [3] 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い，NN の入力とする．NN の出力は，動作の様々なパターンを 3.2 節に示した時系列対応 AAM にて作成し，時刻 s_1,s_2,s_3 の 3 つの動作を合わせて 12 次元としたベクトルとする．また，中間層のノード数は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．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 2 に示す結果になった．

表 2: 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
	たっぷり	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

表 2 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 $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ 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 2 の結果より，動作表現毎の曖昧表現の動作生成結果は，「ザクザク」「混ぜる」を除いては，訓練データで用いた意味ベクトルと未知の意味ベクトルとの差は小さい．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．

誤差最大の「さっと」「ふるう」の予想動作と実験結果動作を，ロボットで生成すると図 3 のようになる．

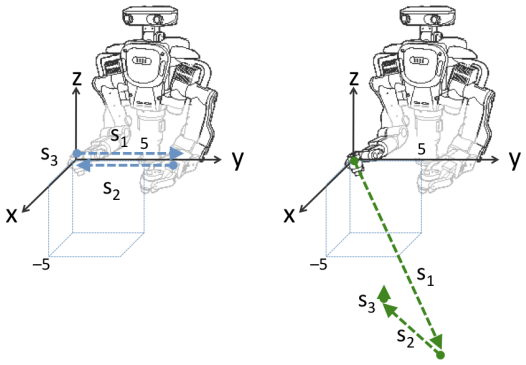

図 3: 「さっと」「ふるう」(左：予想，右：実験結果)

図 3 において，s_1 の動きで実験結果では $\boldsymbol{z}$ 軸マイナス方向へと動いている．s_2 では，$\boldsymbol{z}$ 軸プラス方向へ動いており，更に $\boldsymbol{y}$ 軸マイナス方向の動きは小さい．s_1，s_2 で $\boldsymbol{z}$ 軸方向の動きが出た理由としては，他の動作で $\boldsymbol{z}$ 軸方向の動きがあるためと考えられる．

6　おわりに

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．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．

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] Omer Levy, Yoav Goldberg, “Neural Word Embedding as Implicit Matrix Factorization”, Neural Information Processing Systems, 2014.

[1] http://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html