Understanding Natural Language Sentences with Word Embedding and Multi-modal Interaction

Junpei Zhong*†, Tetsuya Ogata*‡, Angelo Cangelosi†, Chenguang Yang§
*National Institute of Advanced Industrial Science and Technology (AIST), Aomi 2-3-26, Tokyo, Japan
Email: joni.zhong@aist.go.jp
†Centre for Robotics and Neural Systems, Plymouth University, Plymouth, UK
‡ Lab for Intelligent Dynamics and Representation, Waseda University, Tokyo, Japan
§ College of Engineering, Swansea University, Swansea, UK

***Abstract*—Understanding and grounding human commands with natural languages have been a fundamental requirement for service robotic applications. Although there have been several attempts toward this goal, the bottleneck still exists to store and process the corpora of natural language in an interaction system. Currently, the neural- and statstical-based (N&S) natural language processing have shown potential to solve this problem. With the availability of large data-sets nowadays, these processing methods are able to extract semantic relationships while parsing a corpus of natural language (NL) text without much human design, compared with the rule-based language processing methods. In this paper, we show that how two N&S based word embedding methods, called Word2vec and GloVe, can be used in natural language understanding as pre-training tools in a multi-modal environment. Together with two different multiple time-scale recurrent neural models, they form hybrid neural language understanding models for a robot manipulation experiment.**

I. Introduction

Extracting reasonable and efficient features from natural language data is one of the essential requirements to do Natural Language Understanding (NLU) for human-robot interaction. Various kinds of methods based on rule-based, statistical-based or neural-based NLP, have been proposed and implemented in robots. However, one critical problem is that such NLP techniques alone can only process text information from one modality, which is contrary to the language learning process of human beings. There is increasing evidence now that human language is embodied and grounded in multiple modalities [1, 2]. In [3], a two-stream theory in speech processing was proposed, which is similar to the two-stream theory in visual system [4]. In this theory, the ventral stream comprehends the sentences in natural language, while the dorsal stream associates the auditory signals with the motor action. In a conceptual network [5], these two systems may be correlated during the speech synthesis. This theory was partly consistent to the neurocognitive studies done by Pulvermüller and colleagues [6, 7] , where they discovered how the motor representation and lexical representation are correlated in different brain areas. Indeed, neuroimaging studies also discovered that a large number of such interconnections across the visual, auditory and motor areas of the brain are connected and formed in a closed loop in the brain network [8]. Besides the scientific finding, from the perspective of robotic engineering, multi-modal information with natural language commands will be beneficial for the robots to understand the meaningful keywords for specific tasks. Using robots testbeds, researchers also demonstrated how the meaningful keywords (e.g. verbs and nouns) grounded in multi-modalities would result in a more robust and efficient human-robot interaction [9, 10].

The aim of this paper is to augment the multiple modalities grounding experiment [10] by the word embedding techniques. With the availability of big data-sets and high performance computing, the neural- and statistical- (N&S) based NLP gained momentum and are increasingly preferred by researchers and engineers. Compared with these two methods, the stand-alone rule-based needs a well-prepared efficient abstraction rules engine, but it becomes difficult to be regularized when it is dealing with much data. On the contrary, the statistical and neural NLP methods are able to scale better with a large data-sets than the rule-based methods. For instance, recently there has been concepts called "word embeddings" to represent the word or sentence by quantifying the semantic similarities in vectors after learning the context in the corpus. To obtain such distributional properties of linguistic items, large samples of data are often needed to satisfy the training requirement of N&S based NLP methods. Correspondingly, there are mainly two models to realise word embeddings: Word2vec [11] and GloVe [12]. Despite their difference, both of them learn the geometrical encodings (vectors) of words from their co-occurrence information in the corpora. These vectors capture similarities between the corresponding semantic units when they appeared in certain context. Therefore, it is able to infer the distributed representation for semantic units in a vector format. Since this format is able to provide logical semantic meanings, the resulting vectors preserve the logical relation, such as "king $-$ man $+$ woman $=$ queen".

With this unique property, we shed light on using word embedding methods with multi-modal interaction. The novel contributions of this paper are:

1) using pre-trained word embedding model on a larger corpus and applying transfer learning in training multi-modal interaction for robots.
2) using word embedding to represent natural language.

We will firstly introduce the word embedding models in the next section. With the word embedding pre-trained, the robot commands can be understood in a multi-modal interaction by two kinds of multiple time-scales recurrent learning models, namely Multiple Time-scale Recurrent Neural Networks (MTRNN) and Multiple Time-scale Gated Recurrent Units (MTGRU) introduced at Sec. 3. The analysis of experiment results will be shown at Sec. 4. Lastly, a discussion about related works and conclusion will be presented.

II. Word-Embedding Models

Aiming at learning geometrical encodings (vectors) of words from their co-occurrence information, both Word2vec and GloVe share similar objectives, although their cost functions are different [13]. In terms of modelling, the difference between these two methods lies in different learning methods: GloVe is a "count-based" statistical model while Word2vec is a neural learning model.

A. GloVe

GloVe counts the word co-occurrences by calculating the ratios of co-occurrence probabilities. To achieve this, a weighted least squares objective matrix J is used to minimise the dot product of the vectors of two words and the logarithm of their number of co-occurrences.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \hat{w}_j + b_i + b_j - logX_{ij})^2 \qquad (1)$$

where X_{ij} represents the times that the word i appears in the context of the word j, w_i and b_i are the embedding word vector and bias of word i, while $\hat{w}_j$ and b_j are the context word vector and bias of word j. f is a weighting function which prevents learning only from extremely common word pairs.

B. Word2vec

On the other hand, Word2vec works as a predictive model to learn their co-occurrence vectors by a 3-layer feed-forward model[1]. In this model, the input vector and the output vector are the target word i and the context words j, respectively. The model learn the correlation between the target word and the context words in a simple way: the model tries to capture the meaningful semantic regularities by this feed-forward training. And the hidden layer, as a "by-product" after training, represents such regularities between pairs of words.

III. Language Understanding by Multiple Time-scale Models

Compared to the conventional language processing models, we hereby examine two RNN models with the "multiple time-scales" (MT) property. Like the CNN (Convolutional Neural Network) which captures the spatial features, the reasons we consider using time constant parameters is to capture features in a temporal domain. Different time-scale constants are able to capture different temporal dynamics of the sequences: a larger time-scale constant τ means the neuron activities change slowly over time compared with those with a smaller τ so that the slow-context neurons store the input signals with longer time-dependence.

A. Multiple Time-scale Recurrent Neural Networks (MTRNN)

In the MTRNN network [14], the learning of each neuron follows the updating rule of classical firing rate models, in which the activity of a neuron is determined by the average firing rate of all the connected neurons. The neuronal activity is also decaying over time following the updating rule of the leaky integrator model. Therefore, assuming the i-th MTRNN neuron has the number of N connections, the current membrane potential status of a neuron can be defined as both by the previous activation as well as the current synaptic inputs:

$$u_{i,t+1} = (1 - \frac{1}{\tau_i})u_{i,t} + \frac{1}{\tau_i}[\sum_{j \in N} w_{i,j}x_{j,t}] \quad (\text{if } t > 0) \qquad (2)$$

where $w_{i,j}$ represents the synaptic weight from the j-th neuron to the i-th neuron, $x_{j,t}$ is the activity of j-th neuron at t-th time-step and τ is the time-scale parameter. As mentioned, one of the features of MTRNN is that a parameter τ is used to determine the decay rate of the neural activity.

During training, we define the error function E by the Kullback-Leibler divergence in this case:

$$E = \sum_t \sum_{i \in O} y^*_{i,t} log(\frac{y^*_{i,t}}{y_{i,t}}) \qquad (3)$$

where $y^*_{i,t}$ is the target signal of the i-th neuron at the t-th time-step, while $y_{i,t}$ is the actual output. During training, the neurons of MTRNN self-optimise their own weight matrices. Also the internal states of the neurons on the context layers are self-organized, based on different time-constants and the training sequences.

B. Multiple Time-scale Gated Recurrent Units (MTGRU)

To avoid the vanished gradient problem during learning long-term dependencies [15], the gating mechanism has widely adopted, such as GRU (Gated Recurrent Units) and LSTM (Long Short-term Memory). Although both GRU and LSTM have gating mechanisms for the recurrent units, compared with the three gates that exist in LSTM, a GRU has only two gates: a reset gate r and an update gate z. The basic idea of using such a gating mechanism to learn long-term dependencies is similar as in a LSTM, but it was reported that fewer number of gates leads to more efficient in training [16]. When the concept of MT is applied in GRU, it has a similar meaning as in MTRNN: it summarises the dynamics with different time scales of the temporal sequences. Compared with GRU, the output of the multiple time-scale gated recurrent units (MTGRU) contains a so-called "time-scale" constant, which controls how the output from previous time steps influences the current output. It is equivalent that this constant is being

[1] The original report of Word2vec claimed that it is a 2-layer network without the input words. Here we regard it as a 3-layer network following the convention of feed-forward network including input vector itself.

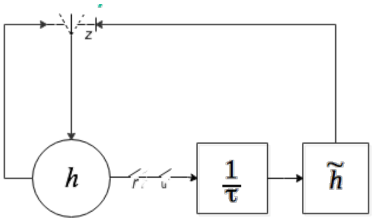


Fig. 1: The MTGRU Unit

multiplied to the output and modulates the mixture of the current and previous states.

Fig. 1 shows internal structure of MTGRU. It demonstrates how the candidate activation $\tilde{h}$ is multiplied with constant $1/\tau$ to the current output. In the meanwhile, the reset gate r_t, update gate z_t, and the candidate activation u_t are computed similarly to those of the original GRU in [17].

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1}) \tag{4}$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1}) \tag{5}$$

$$u_t = tanh(W_{xu}x_t + W_{hu}(r_t \odot h_{t-1})) \tag{6}$$

$$h_t = ((1 - z_t)h_{t-1} + z_t u_t)\frac{1}{\tau} + (1 - \frac{1}{\tau})h_{t-1} \tag{7}$$

Similar as the MTRNN, the pre-defined time-scale τ is introduced to the activation term h_t at Eq. 7 to control the levels abstraction. They controls in what ratio the current and past output to the GRU cell are mixed to compute. A larger τ indicates the past activations have larger influences to the current activation, presenting the long term dynamic feature of the temporal sequences.

In the first MTGRU report [18], the learning formula of the MTGRU and the performances of MTGRU in abstraction was presented. In [19], we also did the first robot manipulation simulation experiment with MTGRU and compared the performances of MTRNN and MTGRU, but based on simple language sentences(one verb and one noun). At the next section, we will augment the word embedding models and investigate the performances of these two word embedding methods and two MT models.

IV. Experimental Results

To examine the networks' performance, we recorded the real world training data from an object manipulation experiment based on an iCub robot [20] (Fig. 3). We used this child-sized humanoid robot with a human instructor teaching the robotic learner a set of language commands whilst providing kinaesthetic demonstration of the named actions. This multi-modal object manipulation experiment was reported in [10]. In the following section, we will use word embedding models as the language input (Fig. 2) and examine the neural activities in these two models. Also additional words will be added to mimic the usage of natural language sentences in the input to examine the performances of the networks. Different from the previous model [10], the language commands are pre-trained with the word embedding models with a relatively large corpus. The Word2vec was run in skip-gram and negative sampling mode. Having 300-dimensional output, both of them are averaged over a sentence. Then the averaged vectors are used as the language inputs of the MTRNN and MTGRU. Both the MTRNN and the MTGRU have the same network parameters when they do the one time-step ahead training (Fig. 2).

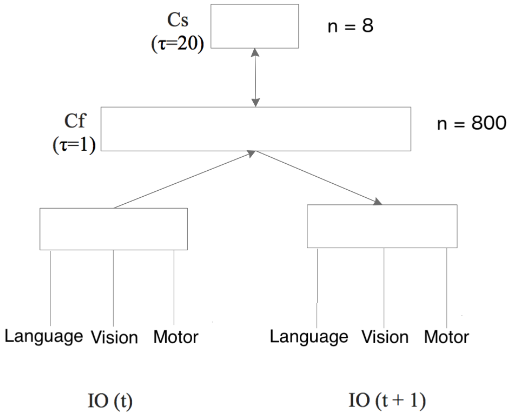


Fig. 2: Language Learning Model based on MTRNN/MTGRU

Fig. 3 shows the setup used in our experiments. The data set was obtained as in [10], following these procedures:

1) Nine objects with significantly different colours and shapes were placed in front of the iCub;
2) A vocal command was spoken by an instructor which includes at least a verb and a noun. For simple commands we used only one verb and one noun in the sentence. For natural language commands other words were added besides of the simple commands. For instance, the simple command "touch ball" was used for training; the natural command "icub touch the ball please" is used for testing.

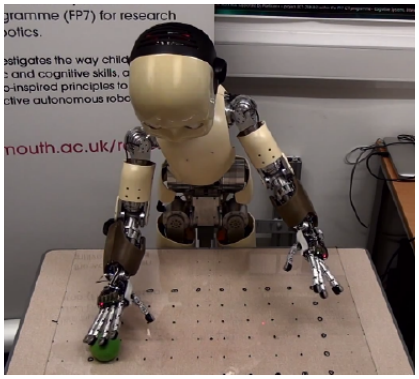


Fig. 3: Experimental Setting

3) The sentence was processed by either Word2vec or GloVe embedding models as the language inputs.
4) With the built-in object tracking, the robot sought for the object and used its neck and head joint angles as visual inputs of the network.
5) The human instructor moved its hand and torso to give an end-to-end training sample of the action according to the command. The trajectories of hand and torso joints were recorded as the motor inputs.

After training with simple command sentences similar as [10], we examined the network performances during testing by adding additional words (we call "noise"). A few benchmarks will be analysed:

1) First of all, the generation error of the trajectories before and after the noise is added will be examined.
2) The representation on the input vector (i.e. the word embedding layer) before and after the noise is added.
3) The comparison of the representations on the context slow (Cs) layers before and after the noise is added.

In the robot experiment, we have 486 training sequences containing language, vision and motor modalities. The Word2vec or GloVe was pre-trained with the English wikipedia corpus (400K words) and each word was converted into a 300-dimensional vector by word embedding. We then converted the whole sentence of language input into a 300-dimensional vector by averaging all the vectors of the words.

A. Generated Trajectories

Firstly, the generated results coming from the natural language commands were examined. These results will be the main consideration of the word embedding methods and will be checked to address whether they are suitable to be implemented in our service robot applications. In the testing mode, we used the "noised" sentences together with the vision inputs for initialisation. Such "noise" contains two words "icub" and "please". The networks generated 100 steps of the motor actions. We then compared the RMS error between the generated results and the original results in Tab. I. The error are based on all the 486 trajectories.

As we can see, the noised error from word embedding models were surprisingly small, compared with the baseline error. To have a more straight-forward comparison, we plot the 6-th trajectories with four models in Fig. 4 (the solid line represents the real value, and the star line represents the generated value). The errors from MTGRU (Figs. 4c & 4d) were larger than MTRNN (Figs. 4a & 4b). The neural representation of Word2vec and GloVe exhibited different dynamics. But the properties with noised inputs were similar: two noise-words drove the networks small perturbations away from the attractors, but they could still generate non-linear dynamics. Note that with large dimension, the MTGRU (~4 days, ~300ms/epoch, NVIDIA TESLA M40) converged slower than the MTRNN (~3 days, ~100ms/epoch, NVIDIA TESLA M40), as the experiment we did [19].

B. Representation of Natural Language Commands

To further examine our hypothesis that the averaging of word embedding vectors has small effects on the representation of the language inputs, we compute the 2-norm between the noised vector and the training vector by the word embedding representation. The 2-norm between two vectors V^a and V^b with dimension n is defined as

$$d(\mathbf{V}^a - \mathbf{V}^b) = \sqrt{\sum_{i=1}^{n}(d_i^a - d_i^b)^2} \qquad (8)$$

Smaller 2-norm implies fewer difference between the simple sentences' vector and the noised sentences' vector. As the previous sub-section, two "noised" words were added: "icub", "please". The average 2-norm of the 81 sentences are shown below in Tab. II.

As we expected, the reason why the generated trajectories were not much affected by the natural language commands was that the added words in the word embedding models did not change a lot in the averaged vectors. Besides, thanks to the robustness of the RNN networks, the errors from the generated trajectories between the simple sentences and the noised sentences did not differ much.

C. Representation of Context Layers

In order to examine the effect from the noised inputs, we further visualised the representation on the context slow (Cs layer in Fig. 2) layers of MTRNN or MTGRU. The slow changing characteristic of this layer showed us how the noised sentence inputs affect the network.

We selected a trajectory sequence (Seq. 5) and observed how the differences of their neural representation by the simple sentence and noised sentences. Due to page restrictions, we only examined the representations in MTRNN. Nevertheless, from the previous experiments about the similar performances of MTRNN and MTGRU, we can expect MTGRU has similar neural activities on the Cs layer. In Fig. 5, we used the initialisation information from the 5-th sequence, and let the network to generate the whole sequence with 100 steps. Then we visualised the slow context layer (τ = 20). Two different kinds of inputs were used in the initialisation: one was the simple input (one verb and one noun) and the other is the noised input (simple inputs plus two words "icub" and "please").

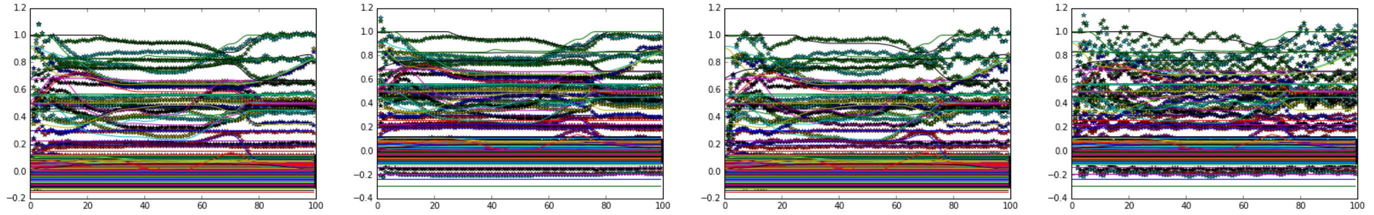
In Fig. 5, we can see that with either Word2vec or GloVe, the slow context (Cs) layer of MTRNN can still keep similar activities, though a few differences appeared. This was probably the reason that with the embedding word models, the MTRNN still resulted in a stable movement generation. Nevertheless, the neural activities with GloVe model were not affected as much as those with the Word2vec model as shown Fig. 5.

V. Related Works

In order to realise language understanding, the robots should be considered in a situated in multi-modal environment as our

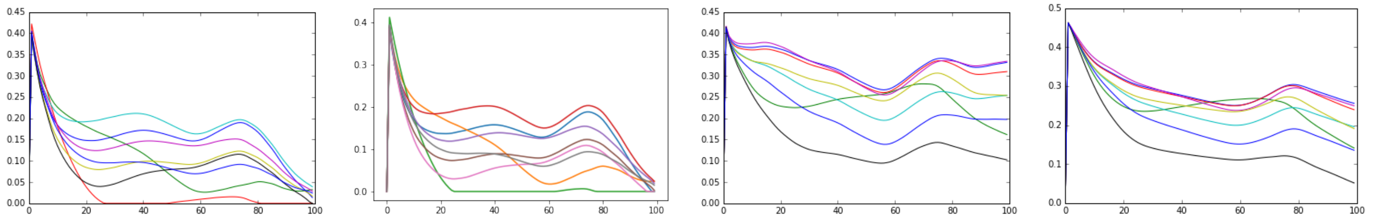
Methods	Word2vec + MTRNN	Word2vec + MTGRU	GloVe + MTRNN	GloVe + MTGRU	MTRNN	MTGRU
RMS	0.063	0.115	0.062	0.114	0.058	0.108

TABLE I: Comparisons of RMSE from Four Models w/ Word Embedding and Baseline w/o Word Embedding



(a) Generated Trajectories by Word2vec + MTRNN (b) Generated Trajectories by GloVe + MTRNN (c) Generated Trajectories by Word2vec + MTGRU (d) Generated Trajectories by GloVe + MTGRU

Fig. 4: Comparisons of the Generated Trajectories



(a) Cs Layer of Word2vec + MTRNN (Simple Sentence) (b) Cs Layer of Word2vec + MTRNN (Noised Sentence) (c) Cs Layer of GloVe + MTRNN (Simple Sentence) (d) Cs Layer of GloVe + MTRNN (Noised Sentence)

Fig. 5: Representation of the Context Layers

Methods	Word2vec	GloVe
2-Norm (“please”)	0.6669	0.5624
2-Norm (“icub”)	0.8721	0.8742
2-Norm (“icub”,”please”)	0.9213	0.9821

TABLE II: Comparisons of 2-Norm from Simple Sentences and Natural Language Sentences based on Word Embedding models

body and brain do. There are various attempts to solve the NLU methods for robotic systems by learning over a corpus of parallel language and multi-modal context data. As the counterpart of the rule-based NLP, several efforts have tried to treat the NLU as a problem of parsing commands into formal meaning representation. But this solution needs more hand-crafting in designing syntactic structure of the multi-modal data: it is non-trivial to use the rule-based methods to parse information to non-atomic information (e.g. dynamic spatial locations [21] or higher-order concepts [22]). Although there have been some attempts in rule-based model to ground multi-modal data with objects and their location relation in images, such as [23]. But again, such non-atomic parsers in an multi-modal environment is difficult to predefined before the robot is actually placed in a domestic environment.

Admittedly, the rule-based approaches have precise processing when they encounter the language fragment which they have been designed for. But the techniques that used in N&S based NLP such as free-form speech recognition, syntactic as well as semantic parsers, make them very suitable for robotic applications while they are able to face a wider range of multi-modal phenomena. Without prior knowledge, [24] learnt a semantic parser about navigation knowledge based on the corresponding verb-argument in the natural language sentence by SVMs. Similarly, based on reinforcement learning, a statistical parser is built to parse specific part of the sentence to the navigation routes [25].

The parsing module of our work is mainly based on recurrent neural processing. But extended from the previous work with a pure neural-based method such as [26], our proposed work falls into a hybrid method between statistical/neural-based NLU on robotic systems. The reason we proposed is that with the recent development of large datasets and the efficient computing power, such a hybrid method would relieve a lot of manual effort but it remains its robustness.

VI. Conclusion

We proposed that using word embedding methods learns distributed word features of natural language commands for robots. Such models are particularly useful and robust to learn the multi-modal information, especially with neural language processing models. We have the following conclusions after studying four model combinations: two word embedding methods (Word2vec & GloVe) and two deep MT recurrent models (MTRNN & MTGRU):

1) in the robot manipulation experiment, using word embedding models makes the robot to "understand" natural language commands with "noised" words;
2) both of the Word2vec and GloVe models provide similar effects to deal with natural language sentences;
3) learning in MTGRU with word embedding is much slower than MTRNN, and it does not provide much improvement in this specific case.

ACKNOWLEDGMENT

The research was partially supported by Waseda SGU Program, the EU FP7 POETICON++ under grant agreement 288382 and the UK EPSRC BABEL.

REFERENCES

[1] A. Cangelosi. "Grounding language in action and perception: from cognitive agents to humanoid robots". In: *Phys Life Rev* 7.2 (2010), pp. 139–151.

[2] L. Steels and M. Hild. *Language grounding in robots*. Springer Science & Business Media, 2012.

[3] G. Hickok and D. Poeppel. "The cortical organization of speech processing". In: *Nature Reviews Neuroscience* 8.5 (2007), pp. 393–402.

[4] M. Mishkin and L. G. Ungerleider. "Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys". In: *Behavioural brain research* 6.1 (1982), pp. 57–77.

[5] J. Zhong, A. Cangelosi, and S. Wermter. "Toward a self-organizing pre-symbolic neural model representing sensorimotor primitives". In: *Front Behav Neurosci* 8 (2014).

[6] F. Pulvermüller and L. Fadiga. "Active perception: sensorimotor circuits as a cortical basis for language". In: *Nature Reviews Neuroscience* 11.5 (2010), pp. 351–360.

[7] F. Pulvermüller et al. "Motor cognition–motor semantics: action perception theory of cognition and communication". In: *Neuropsychologia* 55 (2014), pp. 71–84.

[8] A. D. Friederici. "The cortical language circuit: from auditory perception to sentence comprehension". In: *Trends in cognitive sciences* 16.5 (2012), pp. 262–268.

[9] Y. Sugita and J. Tani. "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes". In: *Adaptive Behavior* 13.1 (2005), pp. 33–52.

[10] J. Zhong et al. "Sensorimotor Input as a Language Generalisation Tool: A Neurorobotics Model for Generation and Generalisation of Noun-Verb Combinations with Sensorimotor Inputs". In: *arXiv preprint arXiv:1605.03261* (2016).

[11] T. Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[12] J. Pennington, R. Socher, and C. D. Manning. "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543.

[13] T. Shi and Z. Liu. "Linking GloVe with word2vec". In: *arXiv preprint arXiv:1411.5595* (2014).

[14] Y. Yamashita and J. Tani. "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment". In: *PLoS Comput Biol* 4.11 (2008), e1000220.

[15] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[16] J. Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[17] K. Cho et al. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014).

[18] M. Kim, M. D. Singh, and M. Lee. "Towards Abstraction from Extraction: Multiple Timescale Gated Recurrent Unit for Summarization". In: *arXiv preprint arXiv:1607.00718* (2016).

[19] J. Zhong, A. Cangelosi, and T. Ogata. "Toward Abstraction from Multi-modal Data: Empirical Studies on Multiple Time-scale Recurrent Models". In: *International Joint Conference on Neural Networks (IJCNN)*. 2017.

[20] G. Metta et al. "The iCub humanoid robot: an open platform for research in embodied cognition". In: *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM. 2008, pp. 50–56.

[21] J. Fasola and M. J. Mataric. "Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 143–150.

[22] C. Matuszek et al. "Learning to parse natural language commands to a robot control system". In: *Experimental Robotics*. Springer. 2013, pp. 403–415.

[23] J. Krishnamurthy and T. Kollar. "Jointly learning to parse and perceive: Connecting natural language to the physical world". In: *Transactions of the Association for Computational Linguistics* 1 (2013), pp. 193–206.

[24] D. L. Chen and R. J. Mooney. "Learning to Interpret Natural Language Navigation Instructions from Observations." In: *AAAI*. Vol. 2. 2011, pp. 1–2.

[25] T. Kollar et al. "Toward understanding natural language directions". In: *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE. 2010, pp. 259–266.

[26] S. Heinrich, S. Magg, and S. Wermter. "Analysing the multiple timescale recurrent neural network for embodied language understanding". In: *Artificial Neural Networks*. Springer, 2015, pp. 149–174.