Robot Language Acquisition Based on Sequence-to-Sequence Learning

Ye Kyaw Thu, Kenta Takabuchi, Kaisei Fukai, Naoto Iwahashi and Takeo Kunishima
Artificial Intelligence Lab., Okayama Prefectural University, Okayama, Japan
{ye, cd27028h, c125043c, iwahashi, kunishi}@c.oka-pu.jp

Abstract

Language acquisition for robot is a challenging topic in the artificial intelligence research area and essential for natural communication between robot and human. In this paper, we proposed language acquisition directly from motion video and user's utterance with multimodal machine learnings without prior knowledge of linguistic or language specific information. Translation between acquired conceptual structure and syllable sequences of a human language (e.g. Japanese language) was carried out by applying machine translation methodologies including sequence-to-sequence learning. Experiments on language acquisition with 500 videos show Encoder-Decoder, Encoder-Decoder with Attention models are able to achieve equal translation performance of baselines that was prepared manually.

1. Introduction

Our research motivation is to develop intelligent robots with the ability to learn natural languages. We believe language acquisition based on multimodal information is the natural and practical approach for robot and human communications. However, there are many challenges such as building language model, classification of objects from image or videos, motion recognition and speech recognition. Nakamura et. al proposed a stochastic model of language and concepts, and knowledge is learnt by estimating the model parameters [26]. In their experiments, generation of object concept formation was done based on visual, haptic, audio and word information. Generally, our robot language acquisition approach of this paper also based on multimodal information similar to [26], except we don't use haptic and word information. Especially, we are focusing on language acquisition without using word information or word class-IDs. Moreover, we applied sequence-to-sequence learning for conversion between conceptual structure to syllable sequences and vice versa. An analysis of the experimental results indicated that conceptual structure learning directly from videos without language specific information is applicable for robot and human communication.

2. Related Work

Language acquisition by robots has been attracting interest in various research fields [33], [2], [43], and several pioneering studies developed algorithms based on inductive learning using sets of pairs, where each pair consists of a word sequence and non-linguistic information about its meaning. In several studies, visual, rather than symbolic, information was given as non-linguistic information [9], [39]. Spoken-word acquisition algorithms based on unsupervised clustering of speech tokens have also been described [11], [15], [34], [27]. Steels examined the socially interactive process of evolving grounded linguistic knowledge shared by communication agents from the viewpoint of game theory and a complex system [37]. In contrast, the method proposed by Iwahashi [16], [17], which is called LCore, focuses on online learning of personally and physically situated language use through verbal and nonverbal interaction with a user in the real physical world. LCore applies information from raw speech and visual observations and tactile reinforcement in an integrated way, and enables a robot to learn incrementally and online beliefs regarding speech units, words, concepts of objects, motions, grammar, and pragmatic and communicative capabilities.

3. Robot Language Acquisition

An overview of our experimental language acquisition system is shown in Figure 3. We used L-Core (refer 3.1) to detect image objects with IDs, trace moving objects, and recognize objects from object manipulation videos. Videos were taken with Microsoft Kinet v1. Feature extraction process is carried out with convolutional neural network (CNN) [8] approach, using CAFFE deep learning framework [19] with IMAGENET (an open trained image network model) [6] from the segmented object images. The extracted features are used for object classification or recognition (refer 3.2). Motion recognition is learned by motion recognition module of L-Core and the method is based on reference-point-dependent Hidden Markov Models (RDP-HMM) [12]. Section 3.3 describes the motion recognition in details. After the motion recognition and object recognition, the conceptual structures of each video are heuristically defined (detail explanation

in Section 3.4). Syllable sequences of Japanese (or) speech information relating to each action is acquired by using the Japanese speech recognition engine Julius [24]. The last step of building robot language acquisition models based on parallel data of conceptual structure (CS) and syllable sequences (SS) of Japanese is carried out by machine translation methodologies.

3.1. L-Core

Although service robots directly interact with people by conversation, most of the systems are trained only with automatic speech recognition engine for specific domain. Our in-house L-Core was designed for language learning from multimodal inputs such as voice, images and motions [17], [28]. The system architecture of L-Core is based on client-server communications and various servers such as vision-server, speech-server, sound-quality-conversion-server, phonetic-typewriter-server, robot-server, Q&A-program are connected with central control program named "ptmove". Recently, it was updated by our laboratory to work with Baxter Research Robot through ROS (Robot Operating System). In this paper, we used L-Core mainly for image segmentation and motion recognition from recorded videos with Microsoft Kinet v1.

3.2. Object Recognition

CAFFE deep learning framework [19] with open trained network model of IMAGENET [6] is used for image features extraction from segmented object images. The extracted features are used for training object recognition. Although we trained several unsupervised classifier such as Complex Tree, KNN, we selected to use highest accuracy classifier Gaussian-kernel SVM [38], [35] for object classification. A SVM is a function that estimates f_x by computing:

$$f(x) = sgn(\sum_i y_i \alpha_i K(x_i, x) + b) \tag{1}$$

where the kernel function $K(x_i, x)$ measures the similarity between the input pattern x and the training sample x_i. The samples x_i for which $\alpha_1, \cdots, \alpha_i \geq 0$ are the support vectors. We used one of the most widely-used kernels, Gaussian and it can be written as follow:

$$K(x, x_i) = e^{\gamma \|x - x_i\|^2} \tag{2}$$

for a given parameter $\gamma > 0$. Here, $\|x - x_i\|^2$ is the Euclidean distance with assumption that similar points are close one to each other. For example, Gaussian kernel will evaluate to 1 if the x and x_i are identical. This assumption is very reasonable in many cases.

The equation 1 express binary classification of SVM approach and we used error-correcting output code multiclass (ECOC) model for classification of ten image object classes [7], [18]. ECOC classification requires a coding design and a decoding scheme. The coding design is for determining the classes that the binary learners train on and the decoding scheme is for determining how the predicted results of the binary classifiers are aggregated. Applying ECOC model possible to improve classification accuracy than other multiclass models [10].

We did 10-fold cross validation with SVM, One-vs-One multiclass classification method and it gives accuracy 98.7% for our ten object classification experiments. The confusion matrix of object classification with Gaussian Kernel SVM on training data can be seen in Figure 2.

3.3. Motion Recognition

While words that refer to objects are nominal, words that refer to motions are relational. The concept of the motion of a moving object can be represented by a time varying spatial relation between a trajector and landmarks, where the trajector is an entity characterized as the figure within a relational profile, and the landmarks are entities characterized as the ground that provide points of reference for locating the trajector [23]. Thus, the concept of the trajectory of an object depends on the landmarks. However, generally, information about what is a landmark is not observed in learning data. The learning method must infer the landmark selected by a user in each scene. Moreover, the type of coordinate system in the space should also be inferred to appropriately represent the graphical model for each concept of a motion [16]. The lexicon containing words referring to objects and motions can be expressed as a probabilistic graphical model (see Figure 3).

Motion recognition of L-Core applied RDP-HMM approach [41]. Let $\mathcal{V}$ denote the observed information. Similar to the above section, $\mathcal{V}$ consists of the trajectory of the moving object, $\mathcal{Y}$, and the set of positions of static objects, $\mathcal{O}$. Motion recognition is formulated as a problem of obtaining the maximum likelihood probabilistic model to output $\mathcal{Y}$.

Let $V = \{v_i | i = 1, 2, ..., |V|\}$ denote a set of learned motion labels, λ_i denote the HMM parameter set which corresponds to motion label v_i, and k_i denote the index of the intrinsic coordinate system of $vi.(\lambda_i, k_i)$ is obtained by using the aforementioned learning method. The maximum likelihood pair of the motion label and reference point indices, $(\hat{i}, \hat{r})$, are searched for as follows:

$$(\hat{i}, \hat{r}) = \underset{i,r}{\operatorname{argmax}} P(\mathcal{Y}|r, vi, \mathbf{R}) \tag{3}$$

$$= \underset{i,r}{\operatorname{argmax}} P(\mathcal{Y}|r, ki, \lambda i, \mathbf{R}) \tag{4}$$

Motion recognition with L-Core achieved 96.8%accuracy on 500 motions. We manually analyzed all 16 error images and found that there are only two error

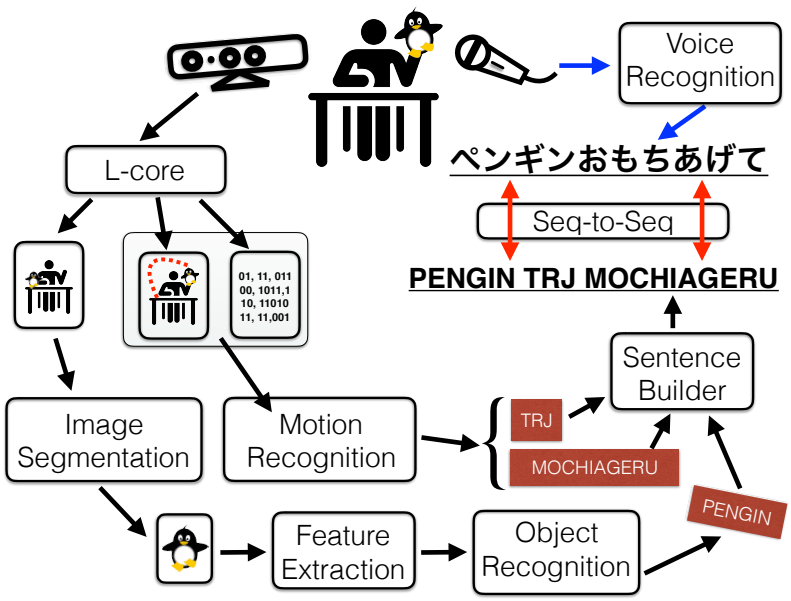


Figure 1. Overview of robot language acquisition (an example with conceptual structure "PENGIN TRJ MOCHIAGERU" and Japanese syllable sequences "ぺんぎんおもちあげて" (Move up penguin))

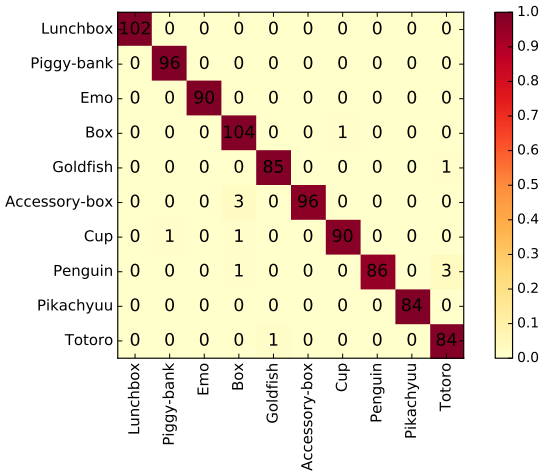


Figure 2. Confusion matrix of object recognition by Gaussian Kernel SVM classification

types. One is error between TRJ and LND recognition (2 images) and another error is between "HANASU" (move-away) and "CHIKAZUKERU" (move-close-to) (15 images).

3.4. Conceptual Structure Generation

Heuristic conceptual structure generation was done by combining Objects IDs (output of the object recognition step) and motion IDs together with TRJ/LND information (output of the motion recognition step).

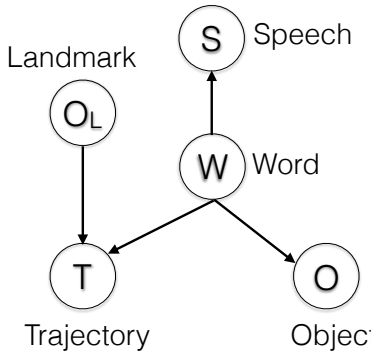


Figure 3. Graphical model of a lexicon containing words referring to objects and motions

This process was handled well by heuristic algorithm because of simple grammar patterns of the conceptual structure and Japanese syllable sequences information from the voice recognition step. In this experiment, there are only seven grammar patterns in total as follows:

1. OBJECT LND OBJECT TRJ HANASU
2. OBJECT TRJ MAWASU
3. OBJECT TRJ MOCHIAGERU
4. OBJECT TRJ OBJECT LND CHIKAZUKERU
5. OBJECT TRJ OBJECT LND HANASU
6. OBJECT TRJ OBJECT LND NOSERU
7. OBJECT TRJ OBJECT LND TOBIKOESASERU

Here, HANASU (move-away), MAWASU (move-circle), MOCHIAGERU (move-up), CHIKAZUKERU (move-close-to), NOSERU (move-onto) and TOBIKOESASERU (move-over) are verbs and always at the end part of the conceptual structures. Two grammar patterns without LND are pattern number 2 and 3 as shown in above list.

3.5. Syllable Sequence Recognition

In this experiment, we used Julius Japanese language ASR engine [24] for speech recognition with one native user for 500 sentences reading. Recording was done inside noiseless environment. The accuracy of syllable sequences recognition with Julius was 88.6% and sentence accuracy was 15.4%. We present syllable recognition errors of five sentences in Table 1. Highlighted and underlined syllables of reference and hypothesis sides are pointing positions of speech recognition errors. We directly used syllable sequences output by Julius ASR engine for experimenting sequence-to-sequence learning with ASR outputs.

4. Methodologies

This section describes four conceptual structure to syllable sequence conversion methodologies used in the experiments.

4.1. Phrase-based statistical machine translation (PBSMT)

A PBSMT translation model is based on phrasal units [22], [30]. Here, a phrase is simply a contiguous sequence of words and generally, not a linguistically motivated phrase. A phrase-based translation model typically gives better translation performance than word-based models. We can describe a simple phrase-based translation model consisting of phrase-pair probabilities extracted from corpus and a basic reordering model, and an algorithm to extract the phrases to build a phrase-table [36]. Figure 4 shows phrase translation entries for conceptual structure phrases "BENTOO LND" (i.e. lunchbox landmark) and "BENTOO TRJ" (i.e. lunchbox trajector) inside a phrase table of PBSMT. In this example, source language is conceptual structure and target language is Japanese syllable sequences.

BENTOO LND ||| べ ん と お ||| 0.310345 0.0274354 . . .
BENTOO LND ||| べ ん と お に ||| 1 0.252406 0.566667 . . .
BENTOO LND ||| ぺ ん と お ||| 0.13171 0.0102134 . . .
BENTOO LND ||| ぺ ん と お に ||| 0.852853 0.0939635 . . .
BENTOO TRJ ||| べ ん と お ||| 0.482759 0.349605 . . .
BENTOO TRJ ||| べ ん と お お ||| 1 0.349605 0.5625 . . .

Figure 4. Some phrase translation entries of PBSMT phrase table

4.2. Hierarchical phrase-based statistical machine translation (HPBSMT)

The hierarchical phrase-based SMT approach is a model [4] based on synchronous context-free grammar. The model is able to be learned from a corpus of un-annotated parallel text. The advantage this technique offers over the phrase-based approach is that the hierarchical structure is able to represent the word re-ordering process. The re-ordering is represented explicitly rather than encoded into a lexicalized re-ordering model (commonly used in purely phrase-based approaches). This makes the approach particularly applicable to language pairs that require long-distance re-ordering during the translation process [1]. Some example of hierarchical phrase-based grammars between conceptual structure and Japanese syllable sequences inside phrase table of HPBSMT are shown in Figure 5. Here, each line in the phrase table represents one translation rule follows by multiple calculated scores for translation process.

BENTOO LND [X] ||| べ ん と お [X] ||| 0.310345 0.0274354 . . .
BENTOO LND [X] ||| べ ん と お に [X] ||| 1 0.252406 0.566667 . . .
BENTOO LND [X] ||| ぺ ん と お [X] ||| 0.126784 0.0102134 . . .
BENTOO LND [X] ||| ぺ ん と お に [X] ||| 0.850649 0.0939635 . . .
BENTOO TRJ [X] ||| べ ん と お [X] ||| 0.482759 0.349605 . . .
BENTOO TRJ [X] ||| べ ん と お お [X] ||| 1 0.349605 0.5625 . . .

Figure 5. Some phrase translation entries of HPBSMT phrase table

4.3. Sequence to Sequence Learning Approaches

4.3.1 Encoder-Decoder Translation Model

Encoder-Decoder translation model is a neural network model that links blocks of LSTMs (Long and Short Term Memory) [13] of source language RNN and target language RNN [5], [42] (see Figure 6(a)). For example, in translation of a source sentence $x_1, x_2, \cdots, x_i, x_{i+1}$ into target sentence $y_1, y_2, \cdots y_i, y_{i+1}$, here, x_1 is a word and x_{i+1} is end of sentence mark "<eos>". Similar to Recurrent Neural Network (RNN) architecture, inside intermediate layer, current context vector e_t of Long Short-Term Memory (LSTM) block pass to next LSTM layer. Although general RNN pass context vector directly from one LSTM block to output layer, source side of RNN does not directly pass to output layer. At the end part of source side RNN, the symbol of end-of-sentence "<eos>" is read.

Inside intermediate layer of target language RNN, it is same as general RNN networks, e_{i+1} is passing directly to next intermediate layer and output y_1 as target word. These output word y_1 will be input word to next LSTM.

Table 1. Some examples of speech recognition errors (Highlighted and underlined characters are pointing position of errors)

Reference	Hypothesis
cho ki n ba ko o ko ˜t pu ni chi ka dzu ke te (ちょ き ん ば こ お こ っ ぷ に ち か づ け て)	cho ki n ba ko o ko pu ni chi ka zu ke te (ちょ き ん ば こ お こ ぷ に ち か ず け て)
e ru mo o ko mo no i re ni no se te (え る も お こ も の い れ に の せ て)	e ru no ko mo no i re ni no se te (え る の こ も の い れ に の せ て)
pi ka chi˜yu u o cho ki n ba ko ka- ra ha na shi te (ぴ か ちゅ う お ちょ き ん ば こ か ら は な し て)	ki ka chi˜yu o cho ki n ba ko ka- ra ha na shi te (き か ちゅ お ちょ き ん ば こ か ら は な し て)
be n to o o pi ka chi˜yu u no u e o to bi ko e sa se te (べ ん と お お ぴ か ちゅ う の う え お と び こ え さ せ て)	be n to o o o pi ka chi˜yu no u yo o to bi ko e sa se te (べ ん と お お お ぴ か ちゅ の う よ お と び こ え さ せ て)
ko mo no i re o ma wa shi te (こ も の い れ お ま わ し て)	ko mo no i re o ma wa su te (こ も の い れ お ま わ す て)

Other steps are same as general RNN, at each time-step, receives an input, updates its hidden state, and makes a prediction. Finally, the target word sequences (i.e. translated target sentence) $y_1, y_2, \cdots y_i, y_{i+1}$ will be generated.

4.3.2 Encoder-Decoder with Attention

As we presented in Section 4.3.1, An Encoder RNN read the entire source sting and generate a fixed length encoded vector to represent the entire source string. The Decoder RNN generate the target string based on the encoded string. Here, compressing input series into one vector is the weak point of Encoder-Decoder model especially for long sentences translation. Encoder-Decoder with *Attention* model was proposed to overcome this problem [25].

The attention model exists between the encoder and the decoder and helps by computing a fixed-size vector that encodes the entire input sequence based on the sequence of *all the outputs* (not only on the last state) generated by the encoder (see Figure 6). In details, the input sequences of encoder $x_1, x_2, \cdots, x_i$ is same with Encoder-Decoder model. However, the output of intermediate layer $\bar{h}_i$ for each x_i is holding globally in Attention model. Basically, decoder of Attention model also same with that of Encoder-Decoder model. $\bar{y}_t$ will be output from input y_t and that $\bar{y}_t$ will become next input as y_{t+1}. However, generation method of $\bar{y}_t$ is different with Encoder-Decoder model.

Let's assume, output of intermediate layer of y_t is h_t. The $\alpha_t(i)$ of following equation will be calculated by using $\bar{h}_i$ that was holding in Encoder side:

$$\alpha_t(i) = \frac{\exp((\bar{h}_i, h_t))}{\sum_{j=1}^{m} \exp((\bar{h}_j, h_t))} \tag{5}$$

Here, $(\bar{h}_i, h_t)$ express inner product of $\bar{h}_i$ and h_t. A variable-length alignment vector $\alpha_t(i)$ is normalization of similarity between y_t and x_i. The context vector c_t was calculated with $\alpha_t(i)$ and $\bar{h}_i$ as follow:

$$c_t = \sum_{i=1}^{m} \alpha_t(i) \bar{h}_i \tag{6}$$

Next, combined vector $[c_t; ht]$ (concatenation vector of source-side context vector c_t and target hidden state h_t) is weighted with linear operator $\boldsymbol{W}_c$ and pass to activation function tanh for getting intermediate layer output $\tilde{h}_t$. An attentional hidden state can be written as follow:

$$\tilde{h}_t = \tanh(\boldsymbol{W}_c[ct; ht]) \tag{7}$$

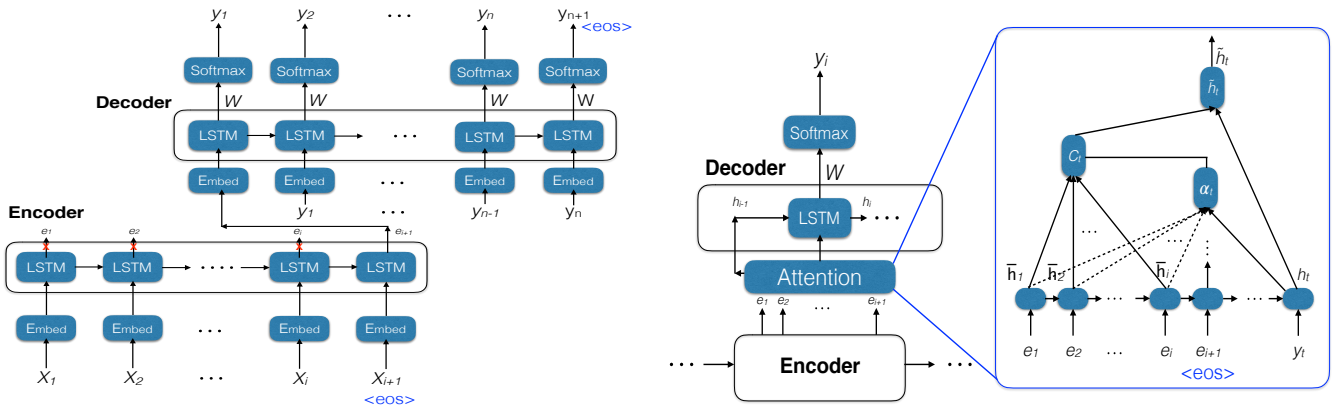
The computation of each weight is using a $softmax$ same as Encoder-Decoder. The $softmax$, as it name mentions, behaves almost like a $argmax$, but it is differentiable. Let's assume that we have an $argmax$ function such that $argmax(x_1, \cdots, x_n) = (0, ..., 0, 1, 0, ..., 0)$ where the only 1 in the output is telling which input is the max . Then, the $softmax$ is defined by $softmax(x_1, \cdots, x_n) = (\frac{e^{x_i}}{\sum_j e^{x_j}})_i$. If one of the x_i is bigger than the other, then $softmax(x_1, \cdots, x_n)$ will be very close to $argmax(x_1, \cdots, x_n)$. For our case, $\bar{y}_t$ will calculated by using $softmax$ function.

5. Experiment

In this section, the setup of the experiments conducted to obtain conceptual structure and syllable sequences from 500 motion videos and user's utterance and language acquisition experiments with machine translation is described.

5.1. Data sets

We prepared five hundred parallel conceptual structure (2,422 words) and syllable sequence of Japanese sentences (7,390 syllables) for baseline. Motion videos for machine learning were taken with Microsoft Kinet



(a) Encoder-Decoder translation model

(b) Encoder-Decoder with attention model

Figure 6. Sequence-to-Sequence learning models

v1 with the setting of six motions (move-close-to, move-away, move-up, move-onto, move-over and move-circle) (see Figure 7) and ten objects (lunch-box, piggy-bank, emo, box, goldfish, accessory box, cup, penguin, Pikachyuu and Totoro) (see Figure 8). Motion videos were taken with a maximum of two objects and speech recognition was done with Japanese ASR engine Julius [24]. Training with four hundred sentences and open testing with one hundred sentences for all experiments (PBSMT, HPBSMT, Encoder-Decoder and Encoder-Decoder-Attention). We trained 1,000 epochs for both Encoder-Decoder and Encoder-Decoder with attention models for all experiments. Data set for experimenting with speech recognition will be contained speech recognition errors and similarly, object recognition errors will be contained in the data set for learning with object recognition outputs.

5.2. Moses SMT system

We used the PBSMT and HPBSMT system provided by the Moses toolkit [20] for training the PBSMT and HPBSMT statistical machine translation systems. The word segmented source language was aligned with the word segmented target languages using GIZA++ [31]. Here, syllable level segmentation was used for both source and target of SS or Japanese language. The alignment was symmetrized by grow-diag-final-and heuristic [21]. The lexicalized reordering model was trained with the msd-bidirectional-fe option [44]. We use SRILM for training the 5-gram language model with interpolated modified Kneser-Ney discounting [40, 3]. Minimum error rate training (MERT) [29] was used to tune the decoder parameters and the decoding was done using the Moses decoder (version 2.1.1) [20]. We used default settings of Moses for all experiments.

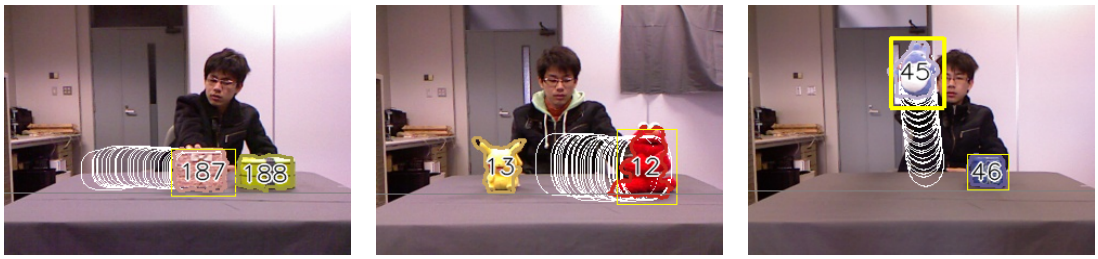
5.3. Framework for RNN

Chainer is a framework for neural network development that provides an easy and straightforward way to implement complex deep learning architectures. A deep learning framework developed by Preferred Infrastructure, Inc. (PFI) (https://preferred.jp/en/) and Preferred Networks, Inc. (PFN) (https://www.preferred-networks.jp/en/). It was released as open source software in June, 2015 (https://github.com/pfnet/chainer). Some key features of Chainer are that it is supported as a Python library (PyPI: Chainer) and is able to run on both CUDA with multi-GPU computers. We used the Chainer Python module (version 1.15.0.1) for the motion to syllable sequence conversion experiments based on RNN trained for 500 epochs.

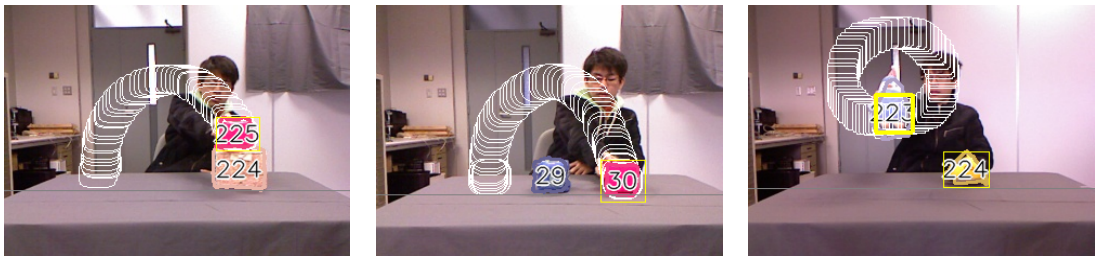
5.4. Evaluation Metrics

We used two automatic criteria for the evaluation of the machine translation output. One was the de facto standard automatic evaluation metric Bilingual Evaluation Understudy (BLEU) [32] and the other was the Rank-based Intuitive Bilingual Evaluation Measure (RIBES) [14].

We used SCLITE (score speech recognition system output) program from the NIST scoring toolkit SCTK version 2.4.10 (http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm) for calculation of Word Error Rate (WER). In our case, WER will be equal to syllable error rate for using syllable segmented Japanese sequences. The SCLITE scoring method for calculating the erroneous words in WER, is as follows: first make an alignment of the G2P hypothesis (the output from the trained model) and the reference (human transcribed) word strings and then perform a global minimization of the Levenshtein distance function which weights the cost of correct words, insertions (I), selections (D) and substitutions (S). The

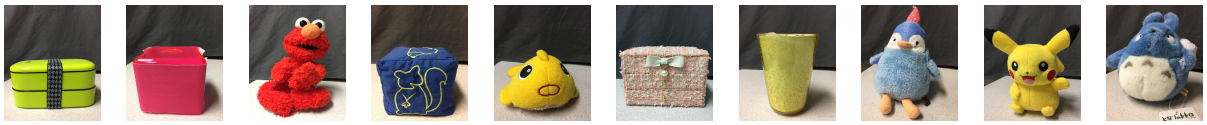


(a) Move-close-to (b) Move-away (c) Move-up



(d) Move-onto (e) Move-over (f) Move-circle

Figure 7. Six motions



(a) Lunch-box (b) Piggy-bank (c) Emo (d) Box (e) Goldfish (f) Accessory-box (g) Cup (h) Penguin (i) Pikachyuu (j) Totoro

Figure 8. Ten objects

formula for WER is as follows:

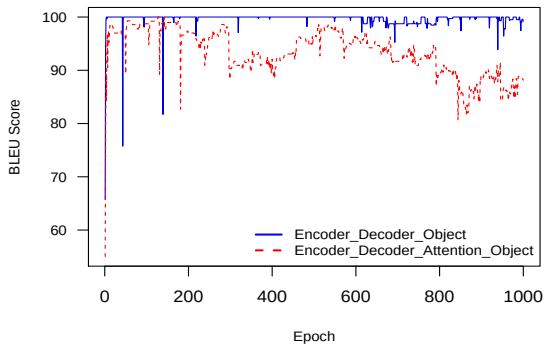
$$WER = (I + D + S)100/N \tag{8}$$

6. Result

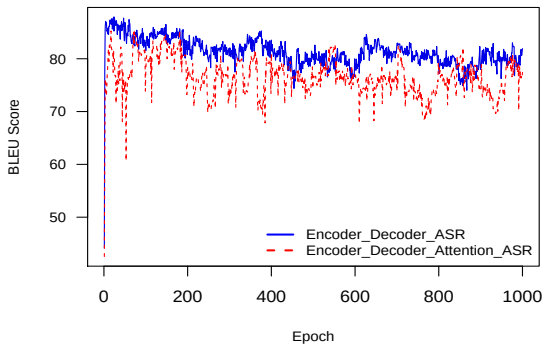
Table 2 and Table 3 show the BLEU and RIBES scores for machine translation between conceptual structure and syllable sequences. The underlined scores indicate the highest scores of the four different approaches. Here, baselines are the results with manually prepared parallel data of conceptual structure and syllable sequences and thus these scores are the best among all. Our target was to reach baseline scores with proposed experiments. Table 2 and Table 3 results indicate Encoder-Decoder achieves best scores for all experiments in terms of both BLEU and RIBES scores. Moreover, Encoder-Decoder and Encoder-Decoder with Attention results for all SS to CS conversion of with object recognition are reached to their baselines. We conduct extensive analysis to better understand our sequence-to-sequence models in terms of learning and the ability to translate together with Object and ASR errors. It is clear to observe in Figure 9, Encoder-Decoder models give better learning curves comparing with Encoder-Decoder with Attention models for current experiments. As we mentioned in Section 3.5, ASR error rate is 88.6% and it is affect the translation performance for both CS to SS and SS to CS (see Table 2 and Table 3).

7. Discussion

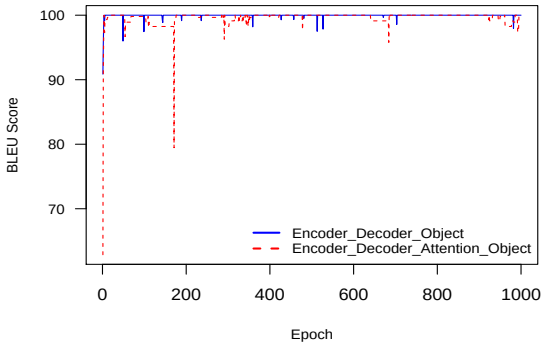
Although grammar patterns of CS and SS are only seven patterns, we considered the differences of average words per sentence between CS and SS. Average words per sentence for CS is 4.83 and for SS is 14.54 and thus we used both BLEU and RIBES score evaluations to measure translation performance [32], [14]. It is pleasant to observe in both Table 2 and Table 3 of SS to CS with object recognition, not only BLEU scores but also RIBES scores are achieved to reach baselines. Moreover, the results of CS to SS with object recognition are also comparable with their baselines. On the other hand, though translation with ASR recognition results also achieved comparable results with their baselines, we are considering the practical issue in language acquisition because of 88.6% of ASR accuracy. As we shown in Figure 11, sequence to sequence learning is possible to learn ASR recognition error sentences



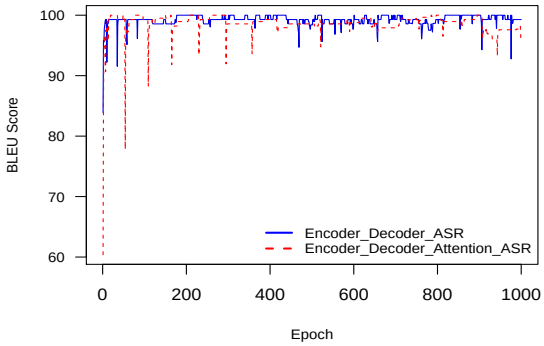
(a) Conceptual structure to Japanese syllable sequence conversion with object recognition results



(b) Conceptual structure to Japanese syllable sequence conversion with ASR results



(c) Japanese syllable sequence to Conceptual structure conversion with object recognition results



(d) Japanese syllable sequence to Conceptual structure conversion with ASR results

Figure 9. Encoder-Decoder and Encoder-Decoder with Attention Results for 1,000 epochs

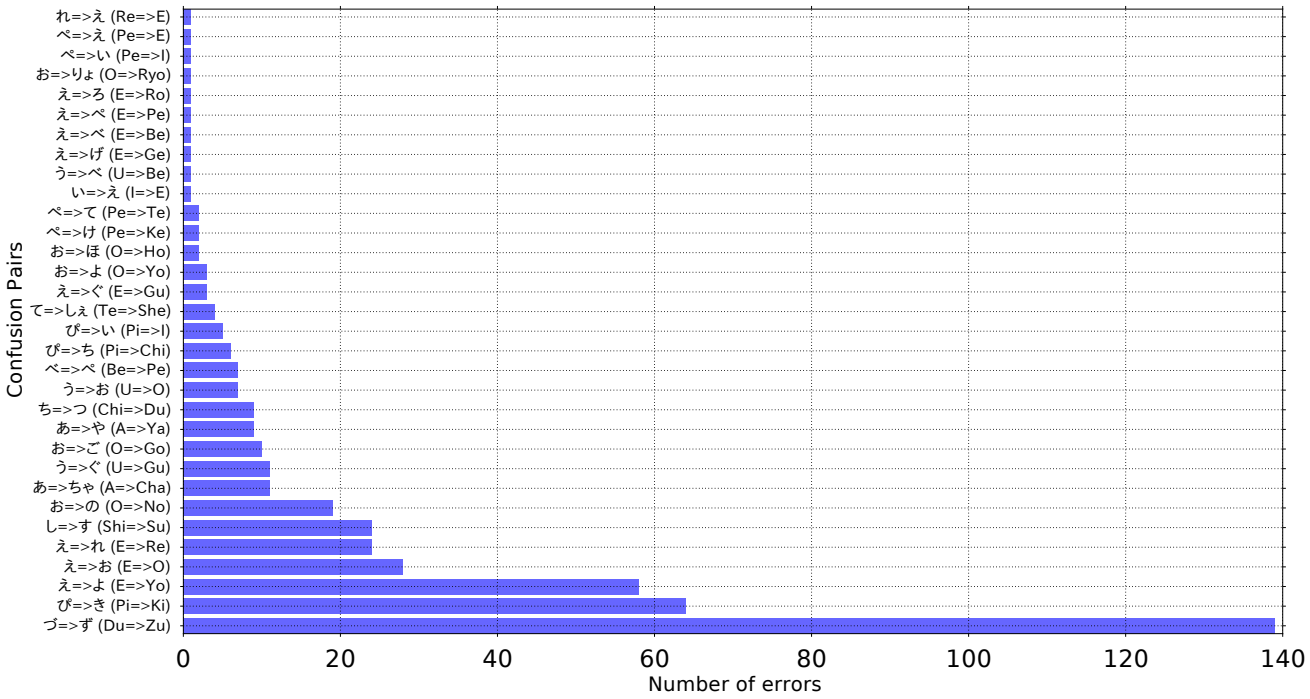


Figure 10. Confusion pairs of automatic speech recognition

Table 2. BLEU scores for machine translation between conceptual structure (CS) and syllable sequences (SS) (**+ ASR Recog** denotes the result with automatic speech recognition, **+ Object Recog** denotes the result with Object recognition)

MT Methods	**CS-to-SS**			**SS-to-CS**		
	Baseline	+ASR Recog	+Object Recog	Baseline	+ASR Recog	+Object Recog
PBSMT	79.68%	71.36%	77.61%	46.69%	46.91%	44.88%
HPBSMT	79.68%	70.71%	77.83%	46.69%	43.50%	45.10%
Encoder-Decoder	100.00%	81.93%	99.22%	100.00%	99.28%	100.00%
Attention	92.53%	77.41%	88.14%	100.00%	95.98%	100.00%

Table 3. RIBES scores for machine translation between conceptual structure (CS) and syllable sequences (SS) (**+ ASR Recog** denotes the result with automatic speech recognition, **+ Object Recog** denotes the result with Object recognition)

MT Methods	**CS-to-SS**			**SS-to-CS**		
	Baseline	+ASR Recog	+Object Recog	Baseline	+ASR Recog	+Object Recog
PBSMT	0.9636%	0.9526%	0.9621%	0.9169%	0.9182%	0.9130%
HPBSMT	0.9636%	0.9477%	0.9618%	0.9169%	0.9176%	0.9137%
Encoder-Decoder	1.0000%	0.9710%	0.9986%	1.0000%	0.9995%	1.0000%
Attention	0.9843%	0.9577%	0.9778%	1.0000%	0.9938%	1.0000%

Figure 11. An example of automatic speech recognition (ASR) error on "Move-up Totoro" (Totoro o mochiagete) sentence together with related image

such as "to to ro o mo tsu a ge te" (Move-up Totoro in English) together with correct CS sentences such as "TOTORO TRJ MOCHIAGERU". However, this kind of learning approach also important in a real world because there is no 100% accurate ASR engine for general domain. We examine on confusion pairs of speech recognition for Japanese SS and found the top ten highest confusion pairs are Du=>Zu, Pi=>Ki, E=>Yo, E=>O, E=>Re, Shi=>Su, O=>No, A=>Cha, U=Gu and O=>Go (see Figure 6). Among them, some of the ASR errors are related to a language nature and one good example from our experiments is the highest confusion pair Du=>Zu. In contrast, we have to consider this kind of language specific ASR recognition errors when we extend experiment for other languages such as English, Chinese, Myanmar.

In this paper, we focused on CS to SS and SS to CS conversion with only object and speech recognition errors and not contained motion recognition errors. This is because motion recognition engine of our in-house L-Core was developed for several years and handle well on motion recognition. As we mentioned in Section 3.3, we conducted motion recognition on 500 videos that is the same data with above experiments and achieved 96.8% accuracy. We plan to combine object, motion and speech recognition errors all together to simulate online language acquisition directly from videos to syllable sequences for next experiments.

8. Conclusion

This paper explore the idea of robot language acquisition research without using language-specific information. Some experimental sequence-to-sequence conversion results between conceptual structure and syllable sequences achieved equal results with manually prepared baseline. We also presented our detail analysis on object and speech recognition errors based on our 500 motion videos and user's utterance. In future work, we would like to extend our experiments with combination of object, motion and speech recognition results for online language acquisition.

9. Acknowledgment

This work was supported by JSPS KAKENHI (grant number 15K00244) and JST CREST ("Symbol Emergence in Robotics for Future Human-Machine Collaboration").

References

[1] F. Braune, A. Gojun, and A. Fraser. Long-distance reordering during search for hierarchical phrase-based smt. In *EAMT 2012: Proceedings of the 16th Annual Conference of the European Association for Machine Translation, Trento, Italy*, pages 177–184. Citeseer, 2012.

[2] A. Cangelosi and M. Schlesinger. *Developmental Robotics: From Babies to Robots*. The MIT Press, 2014.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[4] D. Chiang. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June 2007.

[5] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH*, 2:263–286, 1995.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[9] M. G. Dyer and V. I. Nenov. Learning language via perceptual/motor experiences. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, pages 400–405. Hillsdale, NJ: Erlbaum, 1993.

[10] J. Fürnkranz. Round robin classification. *J. Mach. Learn. Res.*, 2:721–747, Mar. 2002.

[11] A. L. Gorin, S. E. Levinson, and A. Sankar. An experiment in spoken language acquisition. *IEEE Transactions on Speech and Audio Processing*, 2(1):224–240, Jan 1994.

[12] T. Haoka and N. Iwahashi. Learning of the reference-point-dependent concepts on movement for language acquisition. *Technical report of IEICE. PRMU*, 100(442):39–46, nov 2000.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[14] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 944–952, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[15] N. Iwahashi. Language acquisition through a human-robot interface by combining speech, visual, and behavioral information. *Inf. Sci. Inf. Comput. Sci.*, 156(1-2):109–121, Nov. 2003.

[16] N. Iwahashi. *Robots That Learn Language: Developmental Approach to Human-Machine Conversations*, pages 143–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[17] N. Iwahashi, K. Sugiura, R. Taguchi, T. Nagai, and T. Taniguchi. Robots that learn to communicate: A developmental approach to personally and physically situated human-robot conversations. In *Dialog with Robots, Papers from the 2010 AAAI Fall Symposium, Arlington, Virginia, USA, November 11-13, 2010*, 2010.

[18] G. James and T. Hastie. The error coding method and picts. *Journal of Computational and Graphical Statistics*, 7(3):377–387, 1998.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[20] P. Koehn and B. Haddow. Edinburgh's Submission to all Tracks of the WMT2009 Shared Task with Reordering and Speed Improvements to Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164, 2009.

[21] P. Koehn, F. J. Och, , and D. Marcu. Statistical phrase-based translation. In *In Proceedings of the Human Language Technology Conference*, Edmonton, Canada, 2003.

[22] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *HLT-NAACL*, 2003.

[23] R. W. Langacker. *Foundations of Cognitive Grammar, Vol. 2: Descriptive Application*. Stanford University Press, Stanford, 1991.

[24] A. Lee, T. Kawahara, and K. Shikano. Julius –an open source realtime large vocabulary recognition engine. In *in EUROSPEECH*, pages 1691–1694, 2001.

[25] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[26] T. Nakamura, T. Nagai, K. Funakoshi, S. Nagasaka, T. Taniguchi, and N. Iwahashi. Mutual learning of an object concept and language model based on MLDA and NPYLM. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 600–607, 2014.

[27] T. Nakamura, T. Nagai, K. Funakoshi, S. Nagasaka, T. Taniguchi, and N. Iwahashi. Mutual learning of an object concept and language model based on mlda and npylm. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 600–607, Sept 2014.

[28] I. Naoto, T. Ryo, S. Komei, F. Kotaro, and N. Mikio. Robots that learn to converse: Developmental approach to situated language processing. In *NCMMSC2009: Proceedings of the 1National Conference on Man-Machine Speech Communication, Lanzhou, China, Aug. 2009*, pages 532–537, 2009.

[29] F. J. Och. Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, 2003.

[30] F. J. Och and D. Marcu. Statistical phrase-based translation. pages 127–133, 2003.

[31] F. J. Och and H. Ney. Improved statistical alignment models. In *ACL00*, pages 440–447, Hong Kong, China, 2000.

[32] K. Papineni, S. Roukos, T. Ward, and W. Zhu. *Bleu: a Method for Automatic Evaluation of Machine Translation*. IBM Research Report rc22176 (w0109022), Thomas J. Watson Research Center, 2001.

[33] D. Roy. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1):170 – 205, 2005.

[34] D. K. Roy and A. P. Pentland. Learning words from sights and sounds: a computational model. *Cognitive Science*, 26(1):113–146, 2002.

[35] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[36] L. Specia. Tutorial, fundamental and new approaches to statistical machine translation. In *International Conference Recent Advances in Natural Language Processing*, 2011.

[37] L. Steels. Evolving grounded communication for robots. *Trends in Cognitive Sciences*, 7(7):308 – 312, 2003.

[38] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[39] K. Stenning. Terry regier, the human semantic potential: Spatial language and constrained connectionism. *Journal of Logic, Language and Information*, 10(2):266–269, 2001.

[40] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, 2002.

[41] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura. Learning, generation and recognition of motions by reference-point-dependent probabilistic models. *Advanced Robotics*, 25(6-7):825–848, 2011.

[42] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[43] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, and H. Asoh. Symbol emergence in robotics: A survey. *CoRR*, abs/1509.08973, 2015.

[44] C. Tillmann. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, pages 101–104, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.