2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Generating Sentence from Motion by using Large-Scale and High-Order N-grams

Yusuke Goutsu, Wataru Takano and Yoshihiko Nakamura

***Abstract*— Motion recognition is an essential technology for social robots in various environments such as homes, offices and shopping center, where the robots are expected to understand human behavior and interact with them. In this paper, we present a system composed of three models: motion language model, natural language model and integration inference model, and achieved to generate sentences from motions using large high-order N-grams. We confirmed not only that using higher-order N-grams improves precision in generating long sentences but also that the computational complexity of the proposed system is almost the same as our previous one. In addition, we improved the precision by aligning the graph structure representing generated sentences into confusion network form. This means that simplifying and compacting word sequences affect the precision of sentence generation.**

I. Introduction

Humans understand the real world through their multi-modal perception. Perception consists of a large amount of continuous data such as images, audio, and actions, but it is encoded into symbols. The symbols make it possible to understand the real world, predict, and associate by lingualization because they use word meanings by Natural Language Processing (NLP). Also, sentences recover data lost by compression during symbolization by grammar as shown by Fig.1. Thus, humans are different from other animals, and the symbolic system such as language underlies human intelligence. Especially, perception of body motion and the action upon environments through symbolization, lingualization, and sentence are required for humans and humanoid robots to understand human behaviors, estimate partner's intensions, and communicate with gesture or language.

In this paper, we aim to generate sentences from observation of human motions. Our previous framework is composed of three modules[1]. The first module organizes the associations between motion symbols and words. The second with NLP represents linguistic structure as word 2-grams. The third searches a sequence of words that is most likely to represent a human motion by using the score of the above two modules. This framework allows humanoid robots to recognize a human motion as multiple sentences, but some problems still remain. One of them is that the size of training sentences is small in the second module. There is another problem that even if the large-sized training data is applied, natural sentence cannot be generated, because language constraints in a long sentence are not taken into account. To solve this, we extend the 2-gram model to N-gram(N=2-6) models by using a large word N-gram dataset. Relations among long distant words in sentences can be extracted, and complicated sentences can be handled. We also reduce computational cost of searching for sentences corresponding to a motion using a 2-step algorithm. In addition, we propose a method to reduce the word error rate by aligning words not to a conventional graph structure but to a Confusion Network (CN) which is applied in the field of speech recognition for NLP structure. These approaches can improve the performance of motion recognition from the aspects of grammatical correction and variability of the sentences.

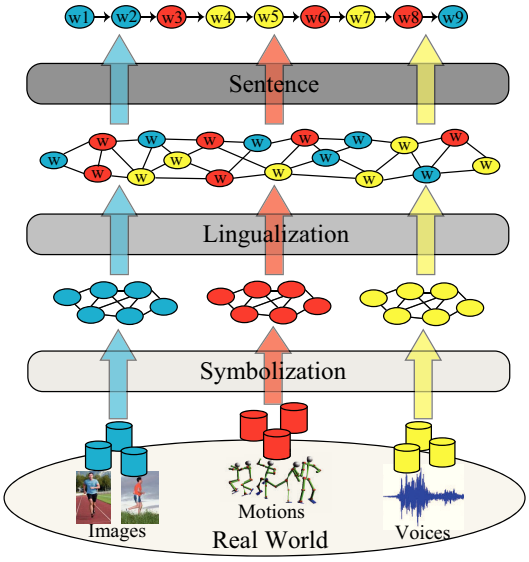

Fig. 1. Humans understand the real world through their multimodal perception. Perception consists of a large amount of continuous data such as images, audio, and actions, but it is encoded into symbols. The symbols make it possible to understand the real world, predict, and associate by lingualization because they use word meanings by NLP. Also, sentences recover data lost by compression during symbolization by grammar.

II. Related Work

A. Symbolization

On the basis of mimesis theory[2] and mirror neurons[3], Ezaki[4] and Inamura[5] proposed a mimesis model. The mimesis model symbolizes motion patterns using imitation learning and conducts motion recognition and generation using motion symbols. In the mimesis model, fullbody motion patterns are represented as time-series signal of multiple joint angles and symbolized into parameters called primitive symbol by Hidden Markov Model (HMM). This framework

The authors are with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan goutsu, takano, nakamura@ynl.t.u-tokyo.ac.jp

has been extended to a sticky HDP HMM and applied to automatic segmentation and symbolization of behavioral patterns[6].

B. Lingualization

Instead of using a statistical model, Sugita[7] and Ogata[8] proposed a bi-directional conversion method by introducing parameters between motion perception and language structure, each of which is represented using Recurrent Neural Network (RNN). As an example of research that uses a statistical model, Takano[9] proposed a translation method between motion symbols and words using the IBM translation model. The model represents the correspondence relationship between time series of motion symbol and strings of words by associative and positional relationship. Hamano[10] also proposed an association method which constructs vector fields of motion symbols and words, and modifies the fields such that a correlation between the two fields can be maximized by a canonical correlation analysis, and derives mappings between the two fields.

C. Sentence

In [11][12], an NLP model represents word sequence from large text corpus by using HMM or CRF. Takano[1] also proposed a motion language model which represents association structure of motion symbols and words, and a natural language model which restricts word sequences stochastically. Given a motion symbol to the motion language model, words are associated from the motion symbol. The words are aligned to make sentences using a word 2-gram model. Additionally, a sentence can be also converted to the corresponding motion symbol. However, this framework cannot generate natural sentences when we use the large-sized training data as pointed out in the above section.

III. Motion Recognition System and Alignment of Graph Structure

In this paper, a motion recognition system consists of three models: "motion language model", "natural language model", and "integration inference model" as shown by Fig.2.

A. Motion Language Model

A motion pattern is symbolized by an HMM, which we will refer to as a motion symbol. The motion symbols are associated with words by a model proposed by Takano[1]. The model consists of three layers of nodes: motion symbols λ, latent states s, and words w. These layers are associated with two kinds of parameters. One is probability $P(s|\lambda)$ that a latent state s is associated with a motion symbol λ. Another is probability $P(w|s)$ that a latent state s generates a word w. The sets of motion symbols, latent states, and words are described by $\{\lambda_i|i = 1, ..., N_\lambda\}$, $\{s_i|i = 1, ..., N_s\}$, $\{w_i|i = 1, ..., N_w\}$ respectively. The k-th training pair, $\{\lambda^k; w_1^k, w_2^k, ..., w_{n_k}^k|k = 1, 2, ..., N\}$, means that the k-th observed motion is recognized as the motion symbol λ^k and that the same motion is manually expressed by the sentence

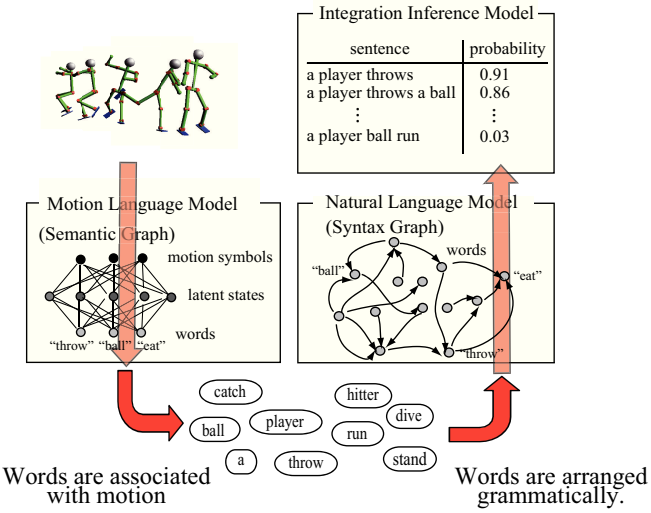

Fig. 2. Overview of interpreting a motion as sentences. The motion language model represents a relationship between motion symbols and words via latent states as a graph structure. The natural language model represents the dynamics of language which means the order of words in sentences. The integration inference model searches for the largest likelihood that sentences are generated from a motion symbol using these model scores.

$\mathbf{w}^{\mathrm{k}} = \{w_1^k, ..., w_{n_k}^k\}$. The model parameters are optimized by EM (Expectation Maximization) algorithm. EM algorithm alternately processes two steps: Expectation step (E-step) and Maximization step (M-step).

E-step calculates distributions of the latent variables based on model parameters estimated in previous M-step. The distributions of the latent variables are provided as follows. □ E-step□

$$P(s|\lambda^k, w_i^k) = \frac{P(w_i^k|s, \lambda, \theta)P(s|\lambda^k, \theta)}{\sum_{j=1}^{N_s} P(w_i^k|s_j, \lambda^k, \theta)P(s_j|\lambda^k, \theta)} \quad (1)$$

where θ is a set of the previously estimated parameters.

M-step estimates the model parameters so as to maximize the summation of expectation of log-likelihood that the symbol of motion pattern λ^k generates the sentence $\mathbf{w}^{\mathrm{k}} = \{w_1^k, ..., w_{n_k}^k\}$. □ M-step□

$$P(s|\lambda) = \frac{\sum_{k=1}^{N}\sum_{i=1}^{n_k} \delta(\lambda, \lambda^k)P(s|\lambda^k, w_i^k)}{\sum_{j=1}^{N_s}\sum_{k=1}^{N}\sum_{i=1}^{n_k} \delta(\lambda, \lambda^k)P(s_j|\lambda^k, w_i^k)} \quad (2)$$

$$P(w|s) = \frac{\sum_{k=1}^{N}\sum_{i=1}^{n_k} \delta(w, w_i^k)P(s|\lambda^k, w_i^k)}{\sum_{j=1}^{N_w}\sum_{k=1}^{N}\sum_{i=1}^{n_k} \delta(w_j, w_i^k)P(s_j|\lambda^k, w_i^k)} \quad (3)$$

where δ is Kronecker delta. The numerators in Eqn.2 and Eqn.3 are the frequency that latent state s is generated from motion symbol λ and the frequency that latent state s is generated from word w respectively. The denominators in Eqn.2 and Eqn.3 are the frequency of motion symbol λ in the training pairs and the frequency of latent state s in the training pairs. We conduct the optimization of model parameters by alternately performing E-step and M-step.

Algorithm 1 finding the maximal word N-gram probability and accumulating backoff weights

```
initialization
repeat
    logP ← find log probability of context from trie node
    if logP is valid then
        record logP as the most specific one found so far
        reset backoffweight
    end if
    if i ≥ maximal context length or context[i] is none vocab then
        break
    end if
    next ← find context[i]
    if next is valid then
        accumulate backoffweight
        set next as next trie node
        increment i
    else
        break
    end if
until break command is occurred
return logP + backoffweight
```

B. Natural Language Model

Many kinds of language models which restrict sentence structures have been proposed in the community of natural language processing. Especially, stochastic models are advantageous because the language model is required to deal with large data. In this paper, we use a word N-gram model because the model can improve the recognition performance easily in addition to simple concept. Word N-gram model is generally represented as an (N-1)-order markov process. In this process, an occurrence probability of i-th word w_i in a word sequence ($\mathbf{w} = \{w_1, w_2, ..., w_n\}$) depends on previous (N-1) words. Thus, word N-gram probability is defined as follows.

$$P(w_i|w_1 w_2 ... w_{i-1}) \simeq P(w_i|w_{i-N+1}, ..., w_{i-1}) \quad (4)$$

In the case of text data, the right side of Eqn.4 can be estimated from relative frequency of words.

$$P(w_i|w_{i-N+1}, ..., w_{i-1}) = \frac{C(w_{i-N+1} ... w_i)}{C(w_{i-N+1} ... w_{i-1})} \quad (5)$$

where $C(w_{i-N+1} ... w_i)$ of Eqn.5 is the frequency of $\{w_{i-N+1} ... w_i\}$. A probability of a word sequence being generated by the natural language model is continuously calculated by summation of the transition probabilities derived in Eqn.5 along the sequence from a start word to an end word. In the case that word N-gram probability cannot be calculated, the back-off weight is added to the word (N-1)-gram probability. The algorithm of calculating the maximal probability including back-off smoothing is shown in Algorithm.1.

C. Integration Inference Model

The process of motion recognition in integration inference model is described as searching for the largest likelihood that sentences are generated from a motion symbol using the motion language model and the natural language model. The likelihood that a sentence $\mathbf{w}$ is generated from a motion symbol λ is derived as

$$P(\mathbf{w}|\lambda) = \prod_{i=1}^{k} P(w_i|\lambda) \cdot \prod_{i=1}^{k} P(w_i|w_{i-N+1}, ..., w_{i-1}) \quad (6)$$

Figure 3 shows how this process conducts step-by-step searching to calculate efficiently in the case of using a large high-order N-gram as natural language model. In the first step, right-to-left search is conducted roughly by using motion language model and simple 2-gram model. In this step, 1-best approximation is conducted by calculating only viterbi paths. The path, a score from each node to goal along

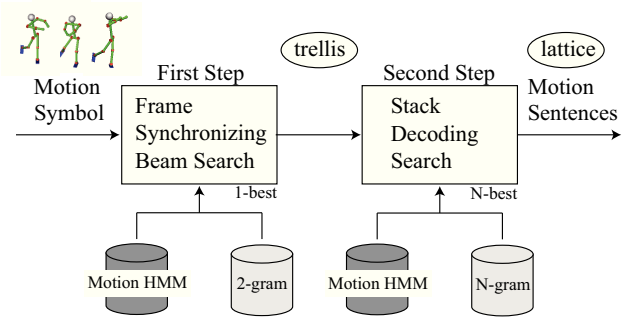

Fig. 3. Outline of the integration inference model. The whole system is divided into 2 steps. The first step is a frame synchronizing beam search using 1-best approximation. The second step is a best-first stack decoding search along the path on the fist step.

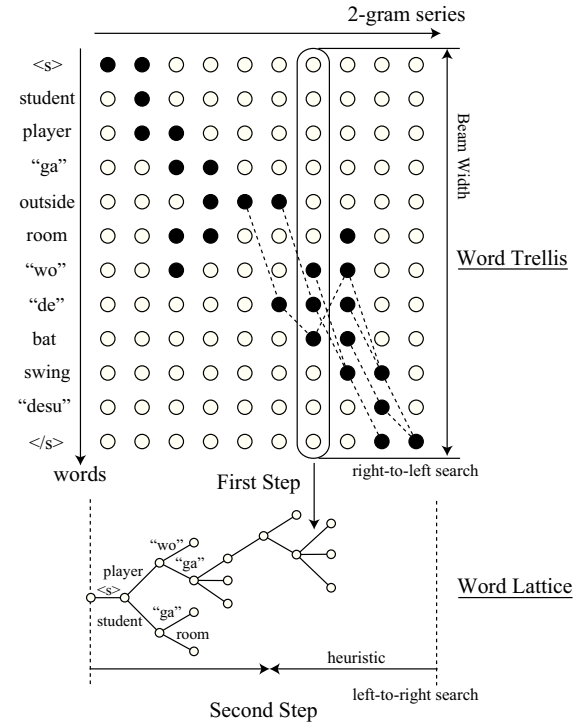

Fig. 4. Word trellis and its use in word expansion on the second step. The word trellis can be available for not only limitation of search path in the second step but also using as heuristic function in A* search algorithm. The word lattice differs from the word trellis in that a word is set on arc with connecting nodes.

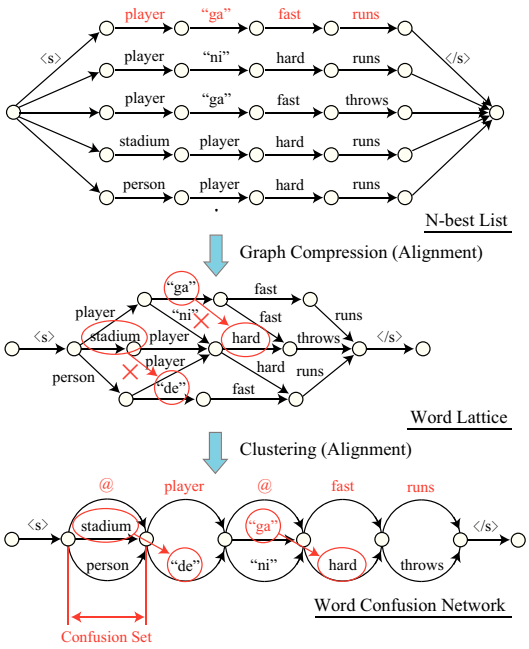

Fig. 5. Process of the alignment from N-best list to word CN. In the word lattice, word sequences are registered in the order of N-best list directly. The word lattice has a defect that input sentences can be matched only the order. The word CN can retrieve word sequences, which are not searched for by the word lattice.

the path, and a word attached to the node are stored in the form of word trellis. The word trellis can be available for not only limitation of search path in second step but also using as heuristic function in A* search algorithm. In the second step, left-to-right search is conducted precisely by using motion language model and up to 6-gram model. In this step, best-first stack decoding search is conducted and searching only necessary path on the first step enables us to recalculate precisely as shown by Fig.4. Because the first generated sentence does not have the largest likelihood necessarily, we also conduct the N-best search that sorts N candidate sentences by its score.

D. Confusion Network

The outputs from integration inference model are not structured in a word lattice but in an N-best list. To simplify and compact the output structure, we align the outputs to word CN. The process of the alignment from N-best list to word CN is as follows.

1) **Graph Compression** We construct a word lattice in order of the N-best list. Transitions between two word can be represented by an arc in the word lattice.
2) **Calculation of Posterior Probability** We calculate posterior probability of each arc in the word lattice by using Forward-backward algorithm.
3) **Clustering** We cluster words on the arc which overlap positionally. Note that a set of positionally-competing words is called Confusion Set (CS).
4) **Addition of "@"(Null Candidate)** In the case that a positionally overlapping word is merged in other class, the sum of posterior probabilities of CS is less than 1. Thus, we add a null candidate which has a posterior probability so that the sum of posterior probabilities becomes one.

In the word lattice, word sequences are registered in the order of N-best list directly. The word lattice has a defect that input sentences can be matched only the order. The word CN can retrieve word sequences, which are not searched for by the word lattice. Note that orders between words in the word lattice can be maintained in the word CN.

TABLE I
CUT-OFF AND TOTAL NUMBER OF EACH N-GRAM

Order	Cut-off	Total Number	
		Before Cut-off	After Cut-off
1-gram	50,000	2,565,424	67,260
2-gram	5,000	80,513,289	3,769,894
3-gram	1,000	394,482,216	17,593,003
4-gram	1,000	707,787,333	20,132,262
5-gram	800	776,378,943	19,485,755
6-gram	500	688,782,933	18,521,684

IV. EXPERIMENTS

A. Dataset

1) Motion Corpus: As training data of motion language model, we use motion corpus which has 467 kinds of motion symbol, each of which several sentences are manually attached to. The total number of sentences is 764. Motion symbol is derived by symbolizing observed motion pattern data which is obtained by measuring 35 marker positions pasted to human body. We used <s> and </s> as sentence beginning and sentence end respectively.

2) Google N-gram Corpus: We use Google N-gram(Web Japanese N-gram) to construct a natural language model. Google N-gram is extracted from Japanese web pages which are crawled by Google. About 20 billion sentences with 255 billion words are targeted at extracted data. The corpus includes 1 to 7-grams which appear more than 20 times. Table I shows the detailed N-gram information.

B. Conditions

1) Motion Language Model: In this experiment, we used 1-gram to construct a word file. In the case that a word is registered in motion corpus but is not in word file, the word is replaced with "<unk>".

2) Natural Language Model: When an N-gram ($N >$ 1) has words that are not registered in the word file, the N-gram is not counted. Thus, if words associated from motion symbol are not registered in the word file, false sentences are probably generated. In this paper, we restricted the order of N-grams up to length 6. If we use Google N-gram as training data without cut-off by frequency, memory required in training process exceeds all available memory. Table I shows the cut-off value which reduces the size of N-grams effectively. In addition, meaningless words such as punctuation characters were observed in many sentences, which resulted in a lower frequency of meaningful words. Therefore, we calculated N-gram probabilities without taking these words into account.

TABLE II

PROCESSING TIME OF TWO MOTION RECOGNITION SYSTEMS

Motion Index		60	260	290	329	386
Words		3	3	3	4	3
Proc Time[s]	Previous	5.90	7.22	7.82	16.80	9.27
	Proposed	7.99	6.97	6.80	49.53	8.03

3) Integration Inference Model: In the process of 2-step searching, we used 2-gram model to construct word trellis. The maximum of log likelihood was calculated as the total score of motion language model and natural language model. We determined that a generated sentence was longer than a sentence attached to the input motion symbol.

C. Evaluation

We used word error rate (WER) to evaluate integration inference model. This is obtained by normalizing edit distance. Note that edit distance is a ratio of substitution, insertion and deletion error which is calculated by using DP matching of words between a generated sentence and a given sentence. Small WER implies that the integration inference model has good performance of sentence generation. The calculations of WER is as follows.

$$\begin{aligned} WER &= \frac{\alpha_S \cdot Sub + \alpha_D \cdot Del + \alpha_I \cdot Ins}{Words} \\ &= Sub_{score} + Del_{score} + Ins_{socre} \end{aligned} \quad (7)$$

where $Words$ is the number of words in a sentence and Sub, Del and Ins are the number of substitution of two words, deletion of a word and insertion of a new word respectively. We gave weights in the ratio of $\alpha_S : \alpha_D : \alpha_I = 2 : 3 : 3$ to Sub, Del and Ins respectively according to the importance of its operation.

V. RESULTS

A. Sentence Generation and Processing Time

The sentence generation by using the proposed system was tested. We arbitrarily selected 5 motion patterns to which sentences composed of 3 or 4 words are given. Figure 6 and Table II show the experimental results of interpreting a motion pattern as sentences. In proposed system, each motion pattern was interpreted as appropriate sentences which are underlined in red as shown by Fig.6. Generated sentences shown in Fig.6 are the top 5 sentences selected from among more than 3-word sentences. Table II shows time required to generate sentences from an input of a motion symbol. In this experiment, we set the order of N-gram and the beam width to 3 and 25 in generation of 3-word sentence, and to 4 and 45 in generation of 4-word sentence respectively.

B. Effect of The Order of N-gram

The effect of the order of N-gram to the processing time and WER was tested. We arbitrarily selected 5 motion patterns to which sentences composed of 5 words are given. Table III shows the average of processing times and WERs in them. Note that WER of each motion pattern is calculated as average of the top 10 sentences in the N-best list. In this experiment, we set the beam width to 70. Increasing the order of N-gram requires more time but WER is decreased.

TABLE III

PROCESSING TIME AND WER WITH CHANGING THE ORDER OF N-GRAM

Order	Proc Time[s]	Sub	Ins	Del	WER	Words
2-gram	7.7	0.29	0.54	0.54	3.84	5
3-gram	56.4	0.22	0.48	0.48	3.33	5
4-gram	227.2	0.14	0.54	0.54	3.50	5
5-gram	417.4	0.43	0.43	0.43	3.44	5
6-gram	618.7	0.36	0.39	0.39	3.06	5

TABLE IV

WER OF 10-BEST AVERAGE, 1-BEST AND CN-BEST

	Sub	Ins	Del	WER	Words
10-best Avg	0.36	0.39	0.39	3.06	5
1-best	0.40	0.32	0.32	2.72	5
CN-best	0.44	0.16	0.16	1.84	5

C. Simplifying or Compacting using The Word CN

The effect of the word CN to the processing time and WER was tested. Table IV shows the average of WERs of the same motion patterns as used in the previous subsection. Note that CN-best means the smallest WER of top words in CSs composed of more than 5 words. In this experiment, we set the order of N-gram and the beam width to 6 and 70 respectively. WER is decreased in the order of 10-best Avg, 1-best, and CN-best.

VI. CONCLUSION

The contributions of this paper are summarized as follows.

1) We proposed a natural language model and a integration inference model in addition to previous motion language model. The natural language model is constructed from large high-order N-grams. The integration inference model searches for sentences with the largest likelihood by using a 2-step searching. By using these models, we achieved to generate natural sentences from motions. Although generation of sentences composed of 3 words by the proposed system consumes 0.12s more than the previous system, it improved the correctness of generated sentences. This means that 2-step searching in the integration inference model is effective for high-speed processing.
2) There was a tendency to improve precision of generating sentences by increasing the order of N-gram. 6-gram model decreased 20.3% of WER compared with 2-gram model as shown by Table III. This is because high-precision language model was constructed by using high-order N-gram.
3) We also proposed an alignment method of graph structure from N-best list to word CN to improve the precision of generating sentences. CN-best decreased 32.4% of WER compared with 1-best as shown by Table IV. This means that simplifying and compacting word sequences on graph structure have effect on the precision of generating sentences.

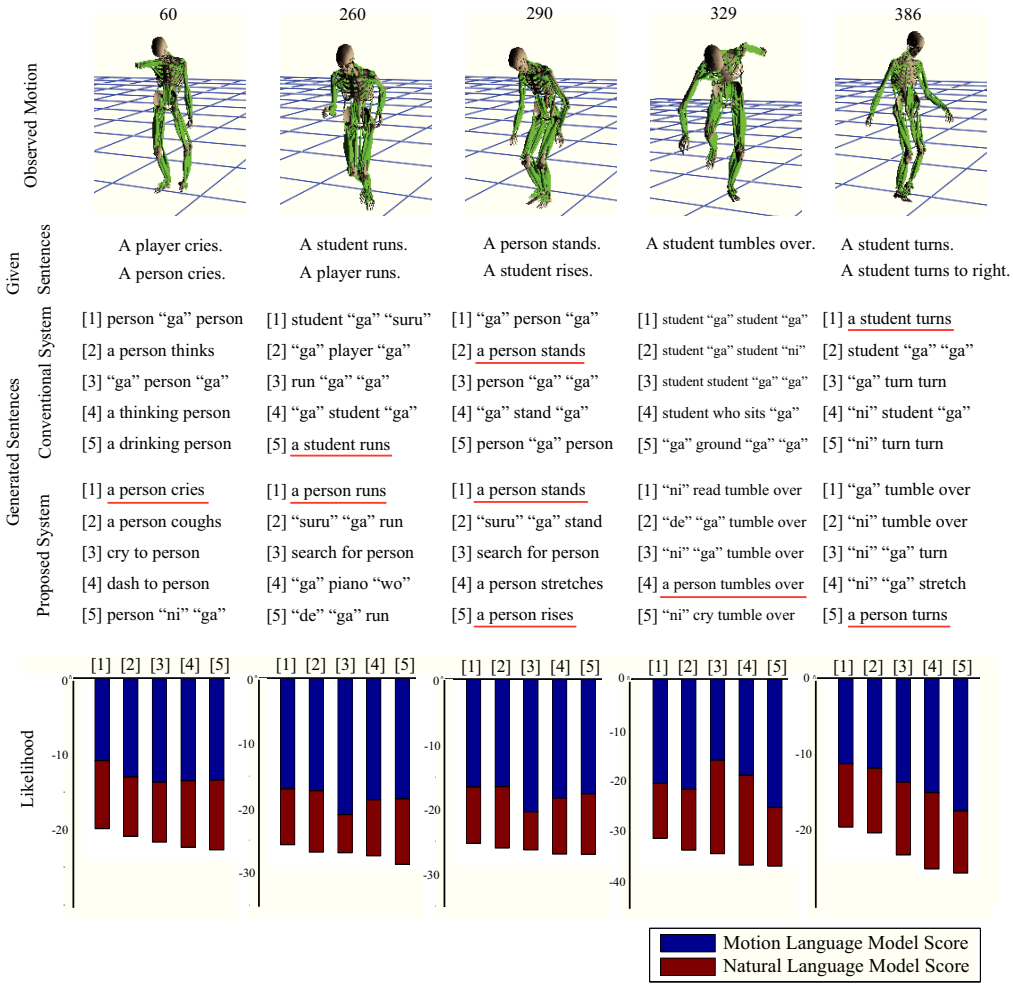

Fig. 6. An observed motion is recognized as a motion symbol. Given sentences are manually assigned to the observed motion. In this figure, two sets of sentences are shown generated by a previous system[1] and a proposed system respectively. For example, A generated sentence of ”a person cries” is the most likely associated with crying motion in ♯60. A generated sentence of ”a person runs” is the most likely associated with running motion in ♯260. A generated sentence of ”a person stands” is the most likely associated with standing motion in ♯290. Japanese particles such as ”ga”, ”wo”, ”ni”, ”de” and ”suru” are unnecessary words. Each likelihood of sentence is represented as a motion language model score and a natural language model score.

ACKNOWKEGEMENT

This research was supported by Grant-in-Aid for Challenging Exploratory Research (24650091), Japan Society for the Promotion of Science.

REFERENCES

[1] W. Takano and Y. Nakamura, “Incremental learning of integrated semiotics based on linguistic and behavioral symbols,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 2545–2550.

[2] M. Donald, *Origins of the modern mind: Three stages in the evolution of culture and cognition*, Harvard Univ Pr, 1991.

[3] G. Rizzolatti, L. Fogassi, and V. Gallese, “Neurophsiological mechanisms underlying the understanding and imitation of action,” *Nature Reviews Neuroscience*, vol. 2, no. 9, pp. 661–670, 2001.

[4] H. Ezaki and Y. Nakamura, “Understanding of other’s motion through self motion primitives,” 1999.

[5] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, “Embodied symbol emergence based on mimesis theory,” *International Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[6] K Takenaka, T Bando, S Nagasaka, T Taniguchi, and K Hitomi, “Contextual scene segmentation of driving behavior based on double articulation analyzer.,” in *Intelligent Robots and Systems, 2012. IROS 2012. IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4847–4852.

[7] 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.

[8] T. Ogata, M. Murase, J. Tani, K. Komatani, and H.G. Okuno, “Two-way translation of compound sentences and arm motions by recurrent neural networks,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 1858–1863.

[9] W. Takano, K. Yamane, and Y. Nakamura, “Capture database through symbolization, recognition and generation of motion patterns,” in *Proc of. IEEE International Conference on Robotics and Automation*, april 2007, pp. 3092–3097.

[10] S. Hamano, W. Takano, and Y. Nakamura, “Motion data retrieval based on statistic correlation between motion symbol space and language,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3401–3406.

[11] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying conditional random fields to japanese morphological analysis,” in *Proc. of EMNLP*, 2004, vol. 2004.

[12] K. Takeuchi and Y. Matsumoto, “Hmm parameter learning for japanese morphological analyzer,” in *Proc, of the lOth Pacific Asia Conference on Language, Information and Computation (PACLING)*, 1995, pp. 163–172.