

---

# Symbol Grounding in Multimodal Sequences using Recurrent Neural Networks

---

**Federico Raue**

University of Kaiserslautern, Germany  
DFKI, Germany  
federico.raue@dfki.de

**Wonmin Byeon**

University of Kaiserslautern, Germany  
DFKI, Germany  
wonmin.byeon@dfki.de

**Thomas M. Breuel**

University of Kaiserslautern, Germany  
tmb@cs.uni-kl.de

**Marcus Liwicki**

University of Kaiserslautern, Germany  
liwicki@cs.uni-kl.de

## Abstract

The problem of how infants learn to associate visual inputs, speech, and internal symbolic representation has long been of interest in Psychology, Neuroscience, and Artificial Intelligence. A priori, both visual inputs and auditory inputs are complex analog signals with a large amount of noise and context, and lacking of any segmentation information. In this paper, we address a simple form of this problem: the association of one visual input and one auditory input with each other. We show that the presented model learns both segmentation, recognition and symbolic representation under two simple assumptions: (1) that a symbolic representation exists, and (2) that two different inputs represent the same symbolic structure. Our approach uses two Long Short-Term Memory (LSTM) networks for multimodal sequence learning and recovers the internal symbolic space using an EM-style algorithm. We compared our model against LSTM in three different multimodal datasets: digit, letter and word recognition. The performance of our model reached similar results to LSTM.

## 1 Introduction

Our brain has an important skill that is to assign semantic concepts to their sensory input signals, such as, visual, auditory. In other words, the sensory inputs can be considered as meaningless physical information and the semantic concepts are linked to their physical features. This scenario can be seen as *Symbol Grounding Problem (SGP)* [1].

Infants in their development ground the semantic concepts to their sensory inputs. For example, several cognitive researchers found a relation between the vocabulary acquisition (audio) and object recognition (visual) [2, 3]. Recently, Asano *et al.* [4] recorded the infant brain activity using three Electroencephalogram (EEG) measures. They found that infants are sensitive to the correspondence between visual stimulus and their sound-symbolic match or mismatch. Furthermore, the lack of one of these components affects the learning behavior, i.e., deafness or blindness [5, 6].

Several models have been proposed for grounding concepts in multimodal scenarios. Yu and Ballard [7] developed a multimodal learning algorithm that maximize the probabilities between spoken words and the visual perception using EM approach. Nakamura *et al.* [8] developed a different approach based on a latent Dirichlet allocation (LDA) for multimodal concepts. They used not only visual and audio information but also haptic information for grounding the concept.

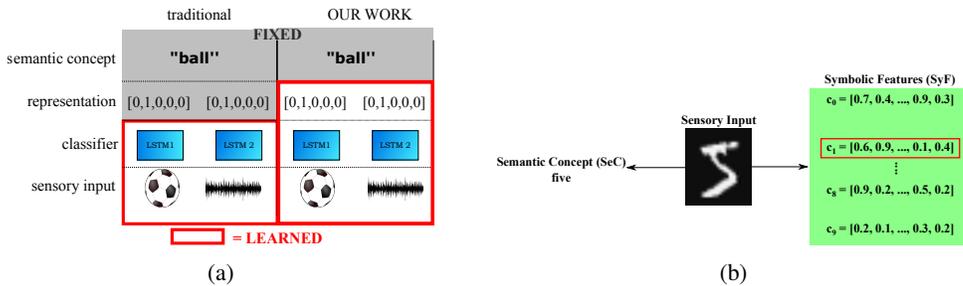


Figure 1: Examples of several components in this work. Figure 1a shows the relation between the traditional approach and our approach for multimodal association. It can be seen that proposed scenario learns the representation of the semantic concept, whereas that relation is fixed in the traditional scenario (red box). Figure 1b illustrates the relation among a *semantic concept*, a visual sensory input and a set of *symbolic features*. In this scenario, there are ten possible options ( $c_0, \dots, c_9$ ) that can be assigned to the concept ‘five’ in order to be represented in the network. In this example, the semantic concept ‘five’ is represented by the symbolic feature  $c_1$ .

Previous work has focused only on segmented inputs. However, recent results in Recurrent Neural Network, mainly Long Short-Term Memory (LSTM), has been successfully applied to scenarios where the input is unsegmented, e.g., OCR and speech recognition. In this paper, we are proposing an alternative solution that exploits those benefits. Furthermore, we address a simplified version of the multimodal symbol grounding: the association of one visual input and auditory input between each other. Our model uses two parallel LSTM networks that segments, classifies and finds the agreement between two multimodal signals of the same semantic sequence. For example, the visual signal is a text line with digit ‘2 4 5’ and the audio signal is ‘two four five’. We want to indicate that our model is trained with less information because the semantic concept and its representation is learned during training. Figure 1a shows the learned components in the traditional scenario and this work. In the traditional scenario, the relation between the semantic concepts and their representation is fixed, whereas that relation in our model is trainable. Moreover, we want to point out that LSTM outputs are used as symbolic features. Figure 1b shows the relation between a semantic concept (*SeC*), a visual sensory input and a set of symbolic features (*SyF*). This relation from now on is called *symbolic structure*. This work is based on Raue *et al.*[9]. In their work, the model was applied to a mono-modal parallel sequence case. In more detail, they learn the association between two text lines, i.e., only visual information. In this work, we explore the model in a more complex scenario where the training is applied on multimodal sequences. Thus, the alignment and the agreement between two modalities are not as smooth as the monomodal scenario.

This paper is organized as follows. Section 2 explains LSTM network as a background information. In Section 3, we describe our model that uses two parallel LSTMs in combination with an EM-based algorithm in order to learn segmentation, classification and symbolic representations. Section 4 explains our experimental setup. Section 5 reports the performance of our model; and, a comparison between our model and a single LSTM network.

## 2 Background: Long Short-Term Memory (LSTM) networks

LSTM was introduced to solve the vanishing gradient in recurrent neural networks [10, 11]. In more detail, the output of the network represents the class probability at each time step. The architecture has already been applied to learn unsegmented inputs using an extra layer called Connectionist Temporal Classification (CTC) for speech recognition [12] and OCR [13]. CTC adds an extra class (called *blank class* ( $b$ )) to the target sequence for learning the monotonic alignment between two sequences. In that case, the alignment is accomplished by learning to insert the blank class at appropriate positions. As a result, LSTM learns the classification and the segmentation. CTC was motivated by the forward-backward algorithm for training Hidden Markov Models (HMM) [14]. In addition, a decoding mechanism extracted the labeled classes from LSTM outputs. Please review the original paper for more details about LSTM and CTC [12].

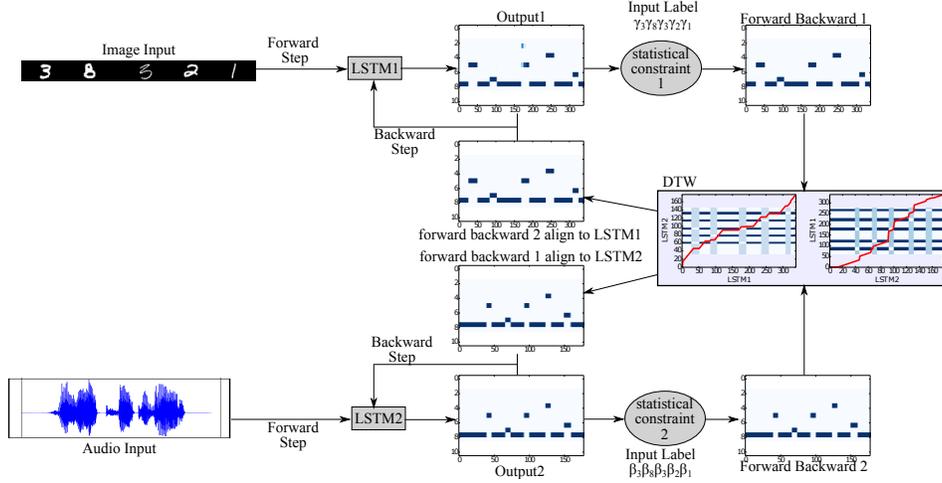


Figure 2: Overview of symbolic association framework. The statistical constraints ( $\gamma$  and  $\beta$ ) guide each LSTM to the internal representation (symbolic feature) for each semantic concept. Also, the monotonic behavior is exploited by DTW. In this manner, LSTM1 output is used as target for LSTM2, and vice versa.

### 3 Multimodal Symbolic Association

As we mentioned in Section 1, the goal of our model is to learn the agreement in a multimodal symbolic sequences. In this case, the term ‘agreement’ is referred to the output classification of both LSTMs are the same (regardless of the modalities). In other words, both LSTMs learn the segmentation, the classification and the symbolic structure in a simplified multimodal scenario.

More formally, we define the multimodal symbolic association problem in the following manner. A multimodal dataset is defined by  $\mathcal{M} = \{(\mathbf{x}_{a,t_1}; \mathbf{x}_{v,t_2}; \mathbf{s}_{1,\dots,n}) | \mathbf{x}_{a,t_1} \in \mathbf{X}_a, \mathbf{x}_{v,t_2} \in \mathbf{X}_v, \mathbf{s}_{1,\dots,n} \in \mathbf{SeC}\}$ .  $\mathbf{X}_a$  and  $\mathbf{X}_v$  are sets of audio and visual sequences, respectively. The length of both sequences can be different.  $\mathbf{s}_{1,\dots,n}$  define the semantic concept sequence of size  $n$  that is represented by two modalities ( $\mathbf{X}_a$  and  $\mathbf{X}_v$ ). As mentioned, the goal is to learn the same symbolic structure that is represented by both modalities. In more detail, each semantic concept is grounded by a similar symbolic feature in both modalities. Also, all semantic concepts are represented by different symbolic features.

In this work, we are proposing a framework that combines two LSTMs for learning a unified symbolic association between two modalities. The intuition behind this idea is to convert from a multimodal input feature space to a common output class space, where two modalities can be associated. Thus, LSTM outputs have the same size. Also, we introduce an EM-training rule based on two constraints: (1) a symbolic representation exists, and (2) two different inputs represent the same symbolic structure. Figure 2 shows a general view of our framework.

In more detail, our model works in the following manner. First, the sequences  $\mathbf{x}_{a,t_1}$  and  $\mathbf{x}_{v,t_2}$  are passed to each LSTM ( $LSTM_1$ ,  $LSTM_2$ ). Then, LSTM outputs ( $\mathbf{z}_{a,t_1}$ ,  $\mathbf{z}_{v,t_2}$ ) and the semantic concept sequence ( $s_1, \dots, s_n$ ) are feeding to the statistical constraint ( $\gamma$  and  $\beta$ ). We want to indicate the LSTM output are used as symbolic feature (SyF). This component selects the most likely relation between semantic concepts and the symbolic features (Section 3.1). As a result, this relation provides information in order to apply the forward-backward algorithm for training (cf. Section 2). Previous steps are independently applied to each LSTM. As we mentioned before, the goal of our model is to learn a unified symbolic structure. With this in mind, the next step in our framework is to align both outputs from the forward-backward algorithm. Our model exploits the monotonic behavior and both sequences are aligned by Dynamic Time Warping (Section 3.2). The aligned output of one LSTM is used as a target of the other LSTM, and vice versa.

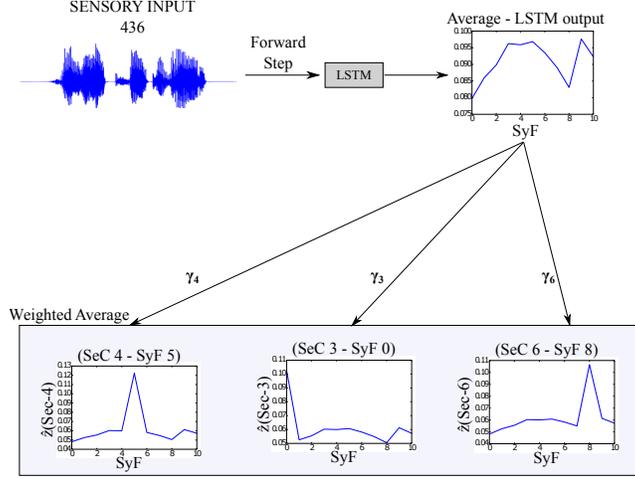


Figure 3: Example of the statistical constraint. The semantic weights ( $\gamma_4, \gamma_3, \gamma_6$ ) modify the average output of LSTM. It can be seen that only one symbolic feature spikes among all for each semantic concept.

### 3.1 Statistical Constraint

The goal of this component is to learn the structure between the *semantic concepts* (*SeC*) and the *symbolic feature* (*SyF*) that are represented by the output of LSTM networks. Our proposed training rule is based on EM algorithm [15]. With this in mind, we define a set of weighted concepts ( $\gamma_1, \dots, \gamma_c$ ) where  $c \in SeC$  and each  $\gamma_c$  is represented by a vector  $\gamma_c = [\gamma_{c,0}, \dots, \gamma_{c,k}]$  where  $k$  is the size of the LSTM output. As a result, the relation can be retrieved by a *winner-take-all* rule. Figure 3 shows an example of the statistical constraint.

The *E-Step* finds the structure between *SeC* and *SyF*. First, we construct the matrix  $\hat{Z}$  which is defined by

$$\hat{Z} = [\hat{z}(1), \dots, \hat{z}(c)]; \quad c \in SeC \quad (1)$$

$$\hat{z}(c) = \frac{1}{T} \sum_{t=1}^T (z_t)^{\gamma(c)}; \quad c \in SeC \quad (2)$$

where  $z_t$  is a column vector that represents LSTM output<sup>1</sup> at time  $t, t \in [1, \dots, T]$  is the timesteps. The column vector  $\hat{z}(c)$  is the weighted average of LSTM output. Next, we convert from matrix  $\hat{Z}$  to matrix  $Z^*$ . A row-column elimination is applied in order to find the symbolic structure for the training. The maximum element ( $i, j$ ) in  $\hat{Z}$  is set to 1 and the elements at the same row  $i$  and column  $j$  are set to 0 except the element ( $i, j$ ). This procedure is repeated  $|SeC|$  times. As a result, only one symbolic representation is selected for each semantic concept.

The *M-Step* updates the set of weighted concepts given the current symbolic structure ( $\hat{Z}$ ). In this case, we are assuming a uniform distribution of semantic concepts. We define the following cost function

$$cost(c) = \left( \hat{z}(c) - \frac{1}{|SeC|} z^*(c) \right)^2; \quad c \in SeC \quad (3)$$

where  $z^*(c)$  is a column-vector of matrix  $z^*$ . The update of  $\gamma(c)$  is accomplished by applying gradient descent

$$\gamma(c) = \gamma(c) - \alpha * \nabla_{\gamma} cost(c); \quad c \in SeC \quad (4)$$

where  $\alpha$  is the learning rate and  $\nabla cost(c)$  is the derivatives of the cost function with respect to  $\gamma(c)$ .

<sup>1</sup>For explanation purposes, the index that represents the modality is dropped, i.e.,  $z_t \equiv z_{a,t_1}$

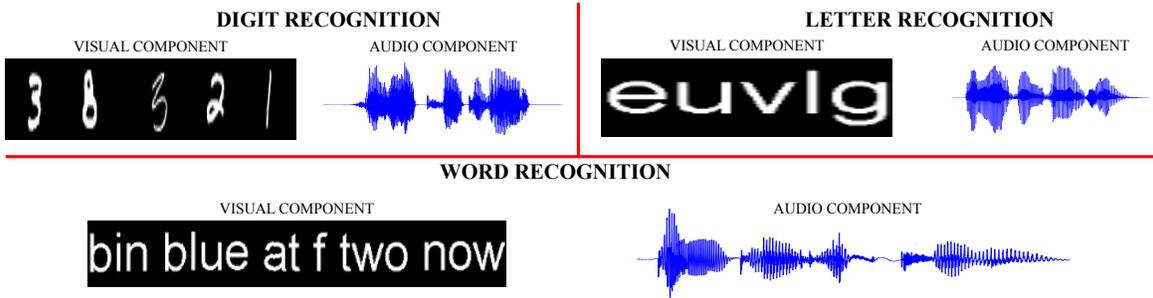


Figure 4: Several examples of the generated multimodal datasets.

After the symbolic structure is learned, the semantic concept is grounded to the symbolic feature, and vice versa. As a result, the semantic concept can be retrieved from the symbolic feature by the maximum element of the following equation:

$$c^* = \arg \max_c z_{k^*}^{\gamma_{c,k^*}} \quad (5)$$

where  $k^*$  is the class decoded from LSTM outputs<sup>2</sup>,  $\gamma_{(c,k^*)}$  is the value at position  $k^*$  in the column vector  $\gamma(c)$ .

### 3.2 Dynamic Time Warping (DTW)

The goal of the second component of our modified learning rule is to align the output of both networks. In other words, the alignment is a mapping function between both networks. Thus, the output of one network can be converted as an approximated output to the other network. This mapping is important for calculating the *error* for updating the weights in the backpropagation step. We apply Dynamic Time Warping (DTW) [16] because of its monotonic behavior scenario. For this purpose, a distance matrix is calculated between each timestep of the forward-backward algorithm from both networks. Equation 6 shows the standard constrains of the path in DTW.

$$DTW[i, j] = dist[i, j] + \min \begin{cases} DTW[i-1, j-1] \\ DTW[i-1, j] \\ DTW[i, j-1] \end{cases} \quad (6)$$

where  $dist[i, j]$  is the distance between the timestep  $i$  of *LSTM1* and the timestep  $j$  of *LSTM2*.

## 4 Experimental Design

### 4.1 Datasets

We generated three multimodal datasets for the following sequence classification scenarios: hand-written digit recognition, printed letter recognition, and word recognition. Each dataset has two components: visual and audio. The visual component is a text line (bitmap) and the audio component is a speech (wav file). Both components represent the same semantic sequence. For example, the semantic sequence ‘3 8 3 2 1’ is represented by a bitmap with those digits and an audio file with ‘three eight three two one’. Figure 4 shows several examples of the multimodal datasets.

**Digit Recognition** The first dataset was generated based on a combination between MNIST [17] and Festival Toolkit [18]. This dataset has ten semantic concepts. Sequences were randomly generated between 3 and 8 digits. The visual component was generated using MNIST dataset. MNIST has already a training set and a testing set. Thus, we kept the same division for creating our raining set and testing set. Each selected digit was attached before and after a random blank background (between 3 and 10 columns). All the selected digits were horizontally stacked. For the audio component, the audio file was generated given the sequences obtained from the visual component and

<sup>2</sup>cf. Section 2

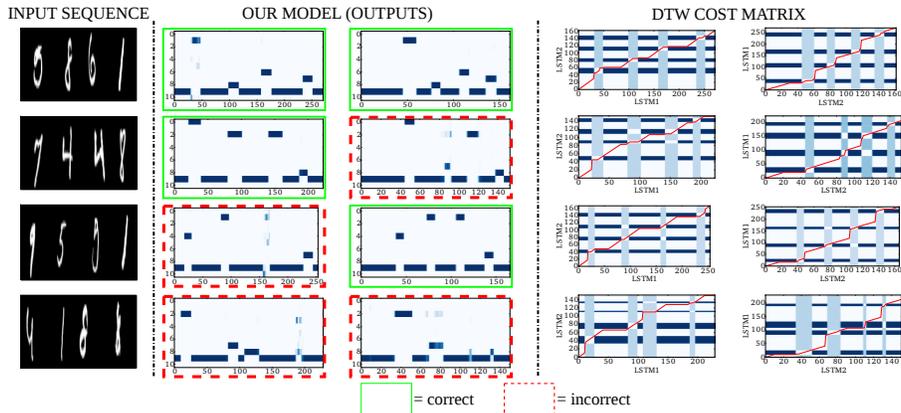


Figure 5: Several examples of the DTW cost matrix. The audio component of the sequences was omitted. The cost matrix (right) shows the path (red line) pass through nine regions. These regions represent the blank class and the semantic concepts.

selected between four artificial voices. As a result, the training set has 50,000 sequences and testing set has 15,000 sequences.

**Letter Recognition** The second dataset was generated following a similar procedure as the previous dataset. This dataset has 27 semantic concepts. We generated text lines of letters as the visual component. The length was randomly selected between 3 and 8 lower characters. The audio component was generated similar to the first dataset using Festival Toolkit. In contrast to MNIST, this dataset does not have an explicit division for the training set and the testing set. Thus, we decided to generate a slightly bigger dataset of 60,000 sequences.

**Word Recognition** The last multimodal dataset is generated based on the audio of the GRID audio-visual sentence corpus [19]. This dataset has 52 semantic concepts. The audio has a fixed sequence length of eight semantic concepts. Also, the audio component is composed by 34 talkers, 18 were males and 16 were females. We generated the text lines of each sequence semantic sequence. The size of this dataset is 34,000 sequences.

## 4.2 Input Features and LSTM Setup

The visual component was used raw-pixel values between 0.0 and 1.0. The audio component was converted to Mel-Frequency Cepstral Coefficient (MFCC) using HTK toolkit<sup>3</sup>. The following parameters were selected for extracting MFCC: a Fourier-transform-based filter-bank with 40 coefficients (plus energy) distributed on a mel-scale, including their first and second temporal derivatives. As a result, the size of the vector was 123. Also, the audio component was normalized to zero mean and unit variance. The training set of the audio component was normalized to zero mean and unit variance.

As a baseline, each component was evaluated using LSTM with CTC layer in order to test the performance of our model. The following parameters were selected for the visual component. The memory size is 20 for the first two datasets and 40 for the last dataset, the learning rate of the network is  $1e-5$  and the momentum is 0.9. The parameters for the audio component are similar but the memory size is 100. The statistical constraint were initialized with 1.0 and the learning rate was set to 0.01 for both networks.

## 5 Results and Discussion

In this paper, the performance of the presented model and the standard LSTM were compared. We want to point out that our goal is not to outperform the standard LSTM, but to know if the

<sup>3</sup><http://htk.eng.cam.ac.uk>

Table 1: Label Error Rate (%) between the standard LSTM and our model. We want to point out that our goal is not to outperform the standard LSTM.

METHOD		DIGITS	LETTERS	WORDS
STANDARD LSTM	VISUAL	3.42 ± 0.84	0.09 ± 0.05	0.45 ± 0.68
	AUDIO	0.08 ± 0.06	1.06 ± 0.14	3.68 ± 0.27
OUR MODEL	VISUAL	2.69 ± 0.55	0.35 ± 0.33	0.51 ± 0.84
	AUDIO	0.15 ± 0.08	1.24 ± 0.50	3.77 ± 0.40

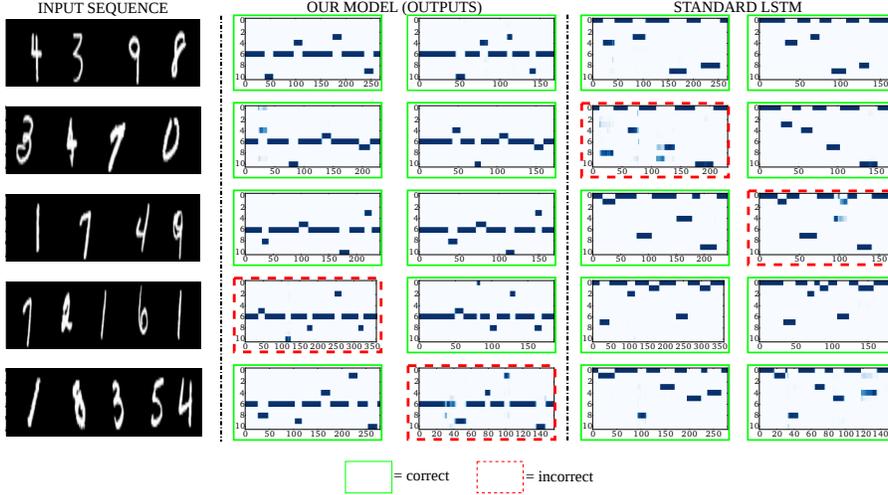


Figure 6: Symbolic Structure between our model and LSTM. The audio component was omitted. Both networks converge to the structure (SyF, SeC): (0, 2), (1, 5), (2, 6), (3, 9), (4, 3), (5, 7), (6, blank-class), (7, 0), (8, 1), (9, 8), (10, 4). It is noted that both models shows similar behavior related to the symbolic structure. LSTM uses a pre-defined structure before the training, whereas the presented model learns the structure during training

performance of our model was in good range. Note that our model has less information than LSTM networks. We randomly selected 10,000 sequences and 3,000 sequences as a training set and testing set (respectively). This random selection was repeated ten times. In the word recognition dataset, we randomly selected 50% male voices and 50% female voices for each training and testing set. We are reporting *Label Error Rate* (LER), which is defined by

$$LER = \frac{1}{|Z|} \sum_{(x,y) \in Z} \frac{ED(x,y)}{|y|} \quad (7)$$

where  $ED$  is the edit distance between the classification of the network  $x$  and the correct output classification  $y$  and  $Z$  is the size of the dataset. Table 1 shows that our model reaches a similar performance to the standard LSTM. In more detail, Fig. 5 shows several examples of the output classification of our model. The first row shows a correct classification of both LSTMs. In this case, both structures of the semantic concepts and the symbolic features are the same. It can be seen that the semantic concept ‘5861’ is represented by the symbolic feature ‘2867’ (dark blue in column 2-3) in both LSTMs. In addition, DTW cost matrix shows an example of the alignment between the both LSTMs. We mentioned in Section 2 that CTC layer adds an extra class. Consequently, our sequence example is converted to ‘b5b8b6b1b’ (nine elements). The DTW cost matrix shows nine regions that the DTW path (red line) crossed. In other words, the alignment happened in the same semantic concept. Furthermore, the alignment still follows the same behavior, even if one or both output classification are wrong.

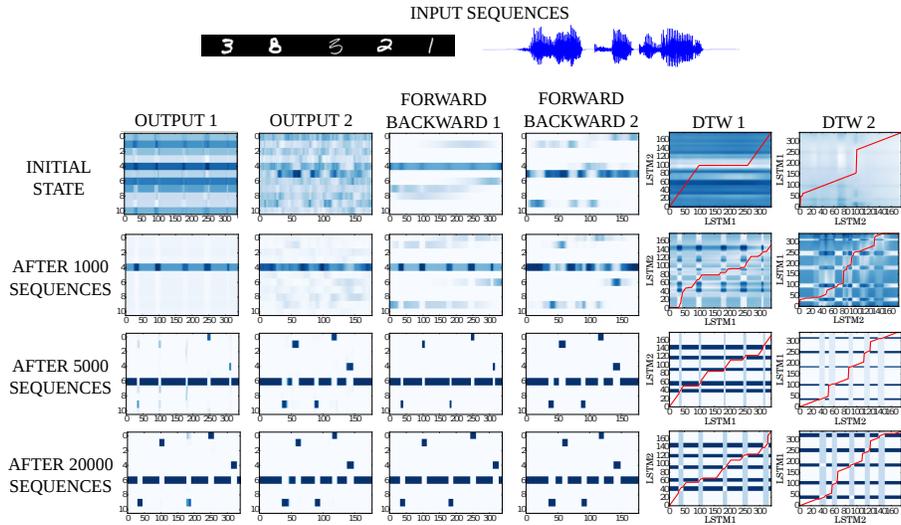


Figure 7: Steps of the training rule. In the beginning, the output of the networks is sparse and they point to align first the blank-class (first three rows). The forward\_backward algorithm shows high values (dark blue) where the blank-class appears. After the blank-class is aligned, the remaining symbolic features are slowly converging to the same representation. The last row shows that both outputs classify the multimodal sequence with similar symbolic features. The DTW cost matrix shows the alignment (red line) between the symbolic features. The alignment has two cases: blank-class to blank-class and semantic concepts to semantic concepts.

Figure 6 shows examples of the symbolic structure. It can be noted that the presented model has a similar behavior as the standard LSTM. For example, our model also learns the blank class for segmenting the semantic concepts. The difference is mainly in the symbolic features for each semantic concept. It is observed that the standard LSTMs used a pre-defined structure between the semantic concepts and the symbolic features. For example, the semantic concept ‘1’ is represented by the symbolic feature ‘1’, the same happened with the rest of semantic concepts. In the other hand, our model learns the structure for each LSTM and both LSTMs converge to a common symbolic structure.

Figure 7 shows the behavior of our model during training. In the beginning, the output of the networks has sparse values and the DTW cost matrices do not have clear regions as in Figure 6. After 10,000 sequences, both networks align first to blank class. DTW cost matrices start showing some initial regions of alignment. After 50,000 sequences, the blank class changes because the structure of the semantic concepts and the symbolic features are not stable. After 20,000 sequences, both networks converge to a common a structure and the DTW cost matrices show a clear DTW path similar to Figure 6.

## 6 Conclusions

This paper has demonstrated that learning symbolic representations of unsegmented sensory inputs is possible with a minimum of assumptions, namely that symbolic representations exist, that two inputs represent the same symbolic content and that classes follow prior distribution. One limitation of our model is the constraint to one-dimension. However, there are many applications in this context, e.g., combining eye tracking system with audio. We will validate our findings with more realistic scenarios, i.e., unknown semantic concepts, aligning a two-dimensional image and a one-dimensional speech, handling missing semantic concepts in one component or both components of the sequence. Finally, it can be seen that this scenario is simple but assigning semantic meanings to symbols is important for language development and remains as an open problem [20, 21, 22].

## References

- [1] S. Harnad, “The symbol grounding problem,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.
- [2] M. T. Balaban and S. R. Waxman, “Do words facilitate object categorization in 9-month-old infants?” *Journal of experimental child psychology*, vol. 64, no. 1, pp. 3–26, Jan. 1997.
- [3] L. Gershkoff-Stowe and L. B. Smith, “Shape and the first hundred nouns.” *Child development*, vol. 75, no. 4, pp. 1098–114, 2004.
- [4] M. Asano, M. Imai, S. Kita, K. Kitajo, H. Okada, and G. Thierry, “Sound symbolism scaffolds language development in preverbal infants,” *cortex*, vol. 63, pp. 196–205, 2015.
- [5] E. S. Andersen, A. Dunlea, and L. Kekelis, “The impact of input: language acquisition in the visually impaired,” *First Language*, vol. 13, no. 37, pp. 23–49, Jan. 1993.
- [6] P. E. Spencer, “Looking without listening: is audition a prerequisite for normal development of visual attention during infancy?” *Journal of deaf studies and deaf education*, vol. 5, no. 4, pp. 291–302, Jan. 2000.
- [7] C. Yu and D. H. Ballard, “A multimodal learning interface for grounding spoken language in sensory perceptions,” *ACM Transactions on Applied Perception (TAP)*, vol. 1, no. 1, pp. 57–80, 2004.
- [8] T. Nakamura, T. Araki, T. Nagai, and N. Iwahashi, “Grounding of word meanings in latent dirichlet allocation-based multimodal concepts,” *Advanced Robotics*, vol. 25, no. 17, pp. 2189–2206, 2011.
- [9] F. Raue, W. Byeon, T. Breuel, and M. Liwicki, “Parallel Sequence Classification using Recurrent Neural Networks and Alignment,” in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*.
- [10] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] S. Hochreiter, “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, Apr. 1998.
- [12] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification,” in *Proceedings of the 23rd international conference on Machine learning - ICML '06*. New York, New York, USA: ACM Press, 2006, pp. 369–376.
- [13] T. Breuel, A. UI-Hasan, M. Al-Azawi, and F. Shafait, “High-performance ocr for printed english and fraktur using lstm networks,” in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 683–687.
- [14] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [15] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society.*, vol. 39, no. 1, pp. 1–38, 1977.
- [16] D. J. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series,” pp. 359–370, 1994.
- [17] Y. Lecun and C. Cortes, “The MNIST database of handwritten digits.”
- [18] P. Taylor, A. W. Black, and R. Caley, “The architecture of the festival speech synthesis system,” 1998.
- [19] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.
- [20] C. J. Needham, P. E. Santos, D. R. Magee, V. Devin, D. C. Hogg, and A. G. Cohn, “Protocols from perceptual observations,” *Artificial Intelligence*, vol. 167, no. 1, pp. 103–136, 2005.
- [21] L. Steels, “The symbol grounding problem has been solved, so whats next ?” *Symbols, Embodiment and Meaning*. Oxford University Press, Oxford, UK, no. 2005, pp. 223–244, 2008.
- [22] S. Coradeschi, A. Loutfi, and B. Wrede, “A short review of symbol grounding in robotic and intelligent systems,” *KI-Künstliche Intelligenz*, vol. 27, no. 2, pp. 129–136, 2013.