Automated Construction of Visual-Linguistic Knowledge via Concept Learning from Cartoon Videos

Jung-Woo Ha, Kyung-Min Kim, and Byoung-Tak Zhang

School of Computer Science and Engineering & Institute for Cognitive Science
Seoul National University, Seoul 151-744, Korea
{jwha, kmkim, btzhang}@bi.snu.ac.kr

Abstract

Learning mutually-grounded vision-language knowledge is a foundational task for cognitive systems and human-level artificial intelligence. Most of knowledge-learning techniques are focused on single modal representations in a static environment with a fixed set of data. Here, we explore an ecologically more-plausible setting by using a stream of cartoon videos to build vision-language concept hierarchies continuously. This approach is motivated by the literature on cognitive development in early childhood. We present the model of deep concept hierarchy (DCH) that enables the progressive abstraction of concept knowledge in multiple levels. We develop a stochastic method for graph construction, i.e. a graph Monte Carlo algorithm, to search efficiently the huge compositional space of the vision-language concepts. The concept hierarchies are built incrementally and can handle concept drift, allowing for being deployed in lifelong learning environments. Using a series of approximately 200 episodes of educational cartoon videos we demonstrate the emergence and evolution of the concept hierarchies as the video stories unfold. We also present the application of the deep concept hierarchies for context-dependent translation between vision and language, i.e. the transcription of a visual scene into text and the generation of visual imagery from text.

Introduction

Recent explosion of data enhances the importance of automatic knowledge acquisition and representation from big data. Linguistically-oriented representation formalisms such as semantic networks (Steyvers and Tenenbaum 2005) and WordNet (Fellbaum 2010) are popular and extremely useful. However, mutually-grounded vision-language concepts are more foundational for cognitive systems that work in perception-action cycles. Existing text-oriented representations are inefficient for learning multimodal concepts from large-scale data, such as videos. Continuous knowledge construction from multimodal data streams is essential for achieving human-level artificial intelligence based on lifelong learning (Muggleton 2014, Zhang 2013).

The task of vision-language learning is to automatically build the relationships between vision and language from multimodal sources of data. Previous works on multimodal learning have focused on either cognitive theory or practical applications. On the practical side, the latent Dirichlet allocation (LDA) models were applied to image annotation (Blei and Jordan 2003) and video object detection (Zhao et al. 2013). Recently, deep learning models were also used for image annotation (Srivastava and Salakutdinov 2012) and descriptive sentence generation (Kiros et al. 2014). However, they mainly focused on automatic annotation rather than constructing semantic knowledge at a higher level. Furthermore, the techniques mostly have concentrated on efficient learning from a static large-scale dataset (Ordornez et al. 2011, Deng et al. 2009) but seldom considered the dynamic change of the contents, i.e. concept drift. Some recent proposals have addressed hierarchical representations (Jia et al. 2013, Lewis and Frank 2013, Abbott et al. 2012), but they are biased to one modality or a static database.

Here we propose a hierarchical model of automatically constructing visual-linguistic knowledge by dynamically learning concepts represented with vision and language from videos, i.e., a deep concept hierarchy (DCH). DCH consists of two or more concept layers and one layer of multiple modalities. The concepts at the higher levels represent more abstract concepts than at the lower layers. The modality layer contains the populations of many microcodes encoding the higher-order relationships among two or more visual and textual variables (Zhang et al. 2012). Each concept layer is represented by a hypergraph (Zhou et al. 2007). This structure coincides with the grounded theory of the human cognition system where a concept is grounded in the modality-specific regions (Kiefer and Barsalou 2013). The structure enables the multiple levels of concepts to be represented by the probability distribution of the visual-textual variables.

The concept construction of DCH from videos involves two technical issues. One is to search a huge space of DCH represented by hypergraphs. The other is to deal with concept drift contained in the video data. For handling these two issues, DCH uses a method based on a Monte Carlo simulation for efficiently exploring the search space, i.e., a graph Monte Carlo (graph MC). The graph MC is a stochastic method for efficiently finding desired graph structures by the repetition of probabilistically generating connections among nodes using observed data instead of sampling. The model structure flexibly grows and shrinks by the graph MC, in contrast to other deep learning models. DCH incrementally learns the concepts by the graph MC and the weight update process while observing new videos, thus robustly tracing concept drift and continuously accumulating new conceptual knowledge. This process is formalized as a sequential Bayesian inference. The learning mechanism is inspired by the cognitive developmental process of children constructing the visually grounded concepts from multimodal stimuli (Meltzoff 1990).

For evaluation, we used the collection of cartoon videos for children, entitled "Pororo", consisting of 183 episodes with 1,232 minutes of playing time. Experimental results show DCH faithfully captures visual-linguistic concepts at multiple abstraction levels, reflecting the concept drift in the progress of the stories. Technically, we investigate the effective combinations of hierarchy architectures and graph MC variants to construct the DCH fast, flexibly, and robustly based on sequentially observed data over an extended period of time. We also present the application of the concept hierarchies for story- and context-aware conversion between the video scenes and the text subtitles.

Visual-Linguistic Concept Representation

To be concrete, we start with the video data from which we extract the vision-language concepts. The whole data set consists of episodes, which are preprocessed into sequences of sentence-image pairs by capturing a scene whenever a subtitle appears. This data generation imitates the process of how a child remembers what the characters say in a scene while observing videos. The vocabulary for the visual words is defined by the set of patches extracted by maximally stable external regions (MSER). Each patch is represented by the vector of SIFT and RGB features. If we represent a textual word as w_i and a visual word as r_i, the utterance-scene is represented as a vector of the form:

$$\mathbf{x}^{(t)} = (\mathbf{w}^{(t)}, \mathbf{r}^{(t)}) = (w_1, ..., w_M, r_1, ..., r_N)\,, \qquad (1)$$

$$D_N = \{(\mathbf{w}^{(t)}, \mathbf{r}^{(t)}) \mid t = 1, ..., T\}\,, \qquad (2)$$

where M and N are the sizes of the textual and visual vocabularies.

Figure 1 shows three instances of the concepts learned from utterance-scene pairs. The objective is to construct a knowledge representation from the data that keeps main conceptual information.

Sparse Population Coding

Sparse population coding (SPC) is a principle to encode data of n variables compactly using multiple subsets of size k. The subset is called a microcode and, typically, its size is small, i.e. $k \ll n$, and thus sparse. The population of microcodes characterizes the empirical distribution of the data in the form of a finite mixture. Previous work shows that SPC is useful for dynamically learning concepts from video data by defining a microcode as a subset of image patches and textual words (Zhang et al. 2012). Formally, the empirical distribution of the observed video data consisting of continuous T scene-utterance pairs can be represented by the population code:

$$P(D \mid \theta) = \prod_{t=1}^{T} P(\mathbf{x}^{(t)} \mid \theta) = \prod_{t=1}^{T} \sum_{i=1}^{M} \alpha_i f_i(\mathbf{w}^{(t)}, \mathbf{r}^{(t)} \mid e_i)\,. \quad (3)$$

where e_i and α_i denote a microcode and its weight, and $f_i(\mathbf{x} \mid e_i)$ is a density function. Also, α_i is a nonnegative value less than 1, summed to be 1. In above equation, a model parameter θ is defined as $\theta = (\boldsymbol{\alpha}, \mathbf{e})$, where $\mathbf{e}$ and $\boldsymbol{\alpha}$

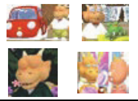
	1~13 episodes (1 DVD)			1~183 episodes (14 DVDs)		
Concepts	Visual nodes	# of nodes (V/L)	Top 15 linguistic nodes	Visual nodes	# of nodes (V/L)	Top 15 linguistic nodes
Pororo		986/230	crong, you, clean, over, draw, huh, to, it, I, up, said, the, moving, is, pororo		12870/1031	crong, you, snowboarding, transforming, rescuing, pororo, the, lamp, seven, are, quack, yellow, not, lollipop, cake,
Eddy		644/198	I, ear, art, midget, game, nothing, say, early, diving, lost, middle, lesson, case, because, snowballs		9008/860	transforming, I, hand, careful, throw, art, suit, midget, farted, reverse, stage, luggage, gorilla, pole, cannon
Tongtong	-	0/0	-		1812/429	kurikuri, doodle, doo, avoid, airplane, crystal, puts, branch, bland, finding, pine, circle, kurikuritongtong, bees, talent

Figure 1: Visual-linguistic representation and development of three character concepts of video contents. A scene-utterance pair is represented by the sets of image patches and words and the concepts of the video stories are represented by these patches and words. *Tongtong* is not seen in episodes 1~13 and appears in episode 56 for the first time.

are the sets of M microcodes and their weights.

Deep Concept Hierarchies

SPC can be considered as a hypergraph, where the hyperedges represent the microcodes. An equivalent representation is a two-layer network where the upper-layer nodes indicate microcodes (hyperedges) and the lower-layer nodes indicate the data variables. Though the representation power is large, the number of upper-layer nodes may grow fast with the growing number of input units, i.e. the visual and textual vocabulary sizes in our video data. To resolve this problem, we introduce additional layers, resulting in a deep concept hierarchy (DCH) shown in Figure 2. A DCH model has explicit concept layers for representing abstract concepts. The connections between layers are sparse, which is contrasted to the deep neural networks that have full connectivity between layers. Also, the number of nodes in each layer flexibly changes for dynamically constructing knowledge as the learning proceeds, which is also contrasted to other deep learning models. This sparse and hierarchical structure reduces the model complexity and DCH pursues a parse modular hierarchical structure, as found in human brains (Quiroga 2012).

Mathematically, DCH represents the empirical distribution of data using a multiple layers of microcodes or concepts. Consider a DCH model with two concept layers in this study. Assume that a node of the top concept layer denotes a character appearing in the video. Let $\mathbf{c}^1=(c_1^1,...,c_{K_1}^1)$ and $\mathbf{c}^2=(c_1^2,...,c_{K_2}^2)$ denote the binary vectors representing the presence of concrete and abstract concepts, where K_1 and K_2 are the sizes of the two vectors. In addition, the sizes of the observable variables, which are M and N in (1), increase whenever observing new words and patches. The probability density of a scene-text pair ($\mathbf{r}$, $\mathbf{w}$) for a given $\mathbf{h}$=($\mathbf{e}$, $\boldsymbol{\alpha}$), $\mathbf{c}^1$, and $\mathbf{c}^2$ can be formulated as

$$P(\mathbf{r},\mathbf{w}\,|\,\mathbf{c}^1,\mathbf{c}^2)=\sum\nolimits_{\mathbf{h}} P(\mathbf{r},\mathbf{w}\,|\,\mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{h}\,|\,\mathbf{c}^1,\mathbf{c}^2)\,, \quad (4)$$

where $\mathbf{e}$ and $\boldsymbol{\alpha}$ denote the population of microcodes and their weights. Each microcode e is defined as two sparse binary vectors whose size is M and N at the time when the scene is observed, respectively. Therefore, DCH can model the concepts as probabilistic associations among words and images. Figure 2 (b) shows an instance of DCH with two concept layers learning concepts from videos.

DCH is basically a hierarchy of hypergraphs. That is, each layer of DCH can be equivalently transformed into a hypergraph by denoting a variable value and a higher-order association as a vertex and a hyperedge, as shown in Figure 2 (c). Now the problem is, the number of possible hyperedges exponentially increases proportional to the number of vertices in a hypergraph. For a k-hypergraph, a hypergraph consisting of hyperedges with k vertices (k-hyperedge) only, the number of possible hypergraphs are $|\Omega|=2^{C(n,k)}$, where $n = |V|$ and $C(n, k)$ denote the number of cases to choose k items from a set with n elements. Ω is denoted as the set of all the hypergraphs. Therefore, the problem space of a DCH model represented by (0, n)-hypergraphs becomes $|\Omega|=2^{2^{|\mathbf{x}|}}$, where $|\mathbf{x}|$ denotes the size of the observable variable set. It is infeasible to explore this huge combinatorial search space with an exhaustive approach.

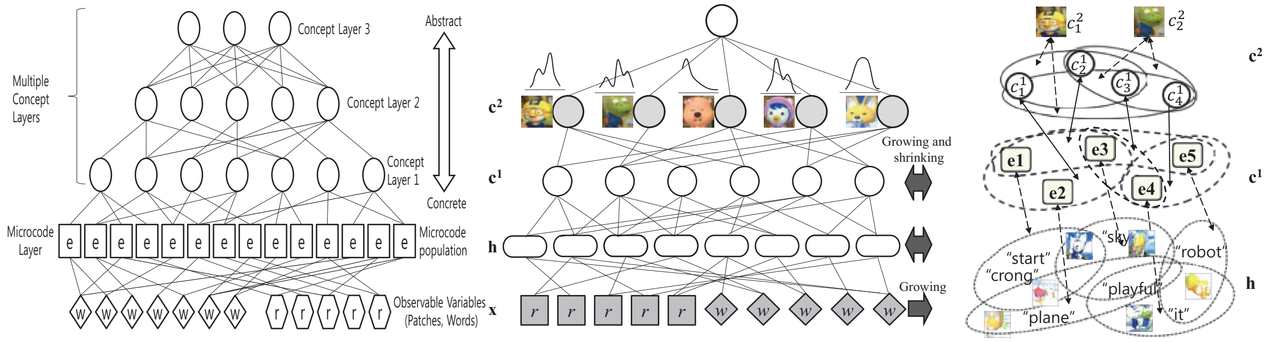
Learning of Deep Concept Hierarchies

Graph Monte Carlo

We propose a method for efficiently constructing hypergraphs incrementally from incoming data. The idea is to use Monte Carlo search on the hypergraph space. The resulting graph Monte Carlo method (graph MC) assumes two conditions:

i) The graph structure in the t-th iteration is determined by that of the t-1 the iteration.
ii) Estimating the empirical distribution asymptotically converges to exploring all theoretical spaces when data are large enough.

Formally, for a given dataset D, an optimal hypergraph



(a) Architecture of deep concept hierarchy (b) Example of deep concept hierarchy learned from Pororo videos (c) Hypergraph representation of (b)

Figure 2: Examples of deep concept hierarchies. (a) presents an architecture of deep concept hierarchy and (b) is an instance of a DCH model with two concept layers learning concepts from Pororo. (c) is the hypergraph representation of (b). Gray boxes in (b) denote observable variables.

G^* corresponding to a model is formulated with Bayes rule:

$$G^* = \arg\max_{G_t} P(G_t \mid D) = \arg\max_{G_t} P(D \mid G_t) P(G_{t-1}), \quad (5)$$

where G_t is a k-hypergraph in the t-th time step. G is constructed to maximize $P(G_t|D)$ by the repetition of replacing hyperedges whenever observing the data:

$$\Delta G = \Delta G \cup \{e\} \text{ and } e = \bigcup_{m=1}^{k} v(x), \quad (6)$$

$$P(e) = \prod_{v \in e} P(v(x)), \quad (7)$$

where ΔG is a new hyperedge set, and e and $v(x)$ denote a generated hyperedge and the vertex corresponding to a variable x. Both the initial values of ΔG and e are empty. $P(e)$ denotes the probability with which e is generated. $P(v(x))$ denotes the probability of which vertices are connected as hyperedges. The graph MC is addressed in terms of the Metropolis-Hastings algorithm under two conditions:

i) A hypergraph G is factorized by its hyperedges to represent a probability distribution (Besag 1974).
ii) ΔG is generated to equivalently represent a sampling instance $\mathbf{x}$.

Then, G^* representing the empirical distribution of the observed data can be constructed by the graph MC. $P(v(x))$ in (7) determines the property of the constructed graph structures, thus playing a role of the learning strategy of the graph MC. Note that $P(v(x))$ is computed from the currently observed data instance according to assumption ii). We define $P(v(x))$ based on three different approaches.

Uniform Graph Monte Carlo

Uniform graph Monte Carlo (UGMC) uses the same probability as $P(v(x))$ for all the variables with the positive value of the data. Then, the probability is defined as follows:

$$P(v(x)) = \left|\{x \mid x \in \mathbf{x}_+^{(n)}\}\right|^{-1} \text{ and } P(e) = C(k, |\mathbf{x}_+^{(n)}|)^{-1}, \quad (8)$$

where $\mathbf{x}_+^{(n)}$ denotes the set of variables with the positive value of the n-th data instance. Then, all the possible hyperedges for a given instance are generated with the same probability.

Poorer-Richer Graph Monte Carlo

The $P(e)$ of each possible hyperedge for a given instance is different from each other in poorer-richer graph Monte Carlo (PRGMC). In PRGMC, a vertex more included in a hypergraph has higher probability. The $P(v(x))$ of PRGMC is defined as follows:

$$P(v(x)) = \frac{R^+\{d(v(x))\}}{|\mathbf{x}|}, \; d(v(x)) = \sum_{e_i \in G_t} \alpha_i h(v(x), e_i), \quad (9)$$

where $R^+(.)$ is a rank function in ascending order, $d(v)$ is the degree of vertex of v, and $h(v, e)$ denotes an indicator function which are 1 when e includes v. For enabling new variables not existing in G_{t-1} to be selected, their $d(v)$ is set to a small value. This approach makes a hypergraph contain the patterns which frequently appear in the training data. Therefore, PRGMC constructs a smaller and denser hypergraph, compared to that built by UGMC.

Fair Graph Monte Carlo

Fair graph Monte Carlo (FGMC) prefers the subpatterns less frequently appearing in the training data, contrary to PRGMC. The $P(v(x))$ is defined as:

$$P(v(x)) = R^-\{d(v(x))\} \big/ |\mathbf{x}|, \quad (10)$$

where $R^-(.)$ is a rank function in descending order. Therefore, a larger and sparser graph is constructed by FGMC and the concepts are represented with much more diverse words and patches.

Learning of Concept Layers

To learning the concept layers we should address three issues: i) determining the number of the nodes of the concrete concept layer $\mathbf{c}^1$ ($\mathbf{c}^1$-nodes), ii) associating between $\mathbf{c}^1$-nodes and modality layer $\mathbf{h}$, and iii) associating between $\mathbf{c}^1$-nodes and the abstract concept nodes ($\mathbf{c}^2$-nodes). The idea is to split the hyperedge set in $\mathbf{h}$ into multiple subgraph clusters, which correspond to the nodes of the $\mathbf{c}^1$ layer. The number of the $\mathbf{c}^1$-nodes are determined based on the distribution of the mean similarities among the hyperedges of a subgraph on all the clusters:

$$Sim(\mathbf{h}^m) = Dist(\mathbf{h}^m) \big/ |\mathbf{h}^m|, \quad (11)$$

where $\mathbf{h}^m$ denotes the subgraph associated with the m-th $\mathbf{c}^1$-node and $Dist(\mathbf{h}^m)$ is the sum of the distance between all the hyperedges of $\mathbf{h}^m$. Then, the distance is estimated by converting the words into the real-value vectors by word2vec (Mikolov et al. 2013). Considering the story-specific semantics of words, we used the corpus of cartoon video subtitles instead of conventional text corpora. If $Sim(\mathbf{h}^m) > \theta_{max}$, $\mathbf{h}^m$ is split into two subgraphs and a new $\mathbf{c}^1$-node is added into the $\mathbf{c}^1$ layer and associated with one of the split subgraph. On the other hand, if all the mean similarities are smaller than θ_{min}, the number is reduced and the associations are conducted again. θ_{max} and θ_{min} are adaptively determined from the mean and the variance of the similarity. $\mathbf{c}^2$-nodes are associated with a $\mathbf{c}^1$-node when the characters corresponding to the $\mathbf{c}^2$-nodes appear in the hyperedges of the subgraph associated with the $\mathbf{c}^1$-node.

Incremental Concept Construction

DCH learns incrementally, i.e. builds the visual-linguistic concepts dynamically while sequentially observing scene-text pairs. We use all the scene-text pairs of one episode as a mini corpus. On sequential observation of the episodes, DCH predicts the concepts from the population and updates the population from the observed data and characters. Formally, this implements a sequential Bayesian estimation:

$$P_t(\mathbf{h},\mathbf{c}^1 \mid \mathbf{r},\mathbf{w},\mathbf{c}^2) = \frac{P(\mathbf{r},\mathbf{w} \mid \mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{c}^2 \mid \mathbf{c}^1,\mathbf{h})P_{t-1}(\mathbf{h},\mathbf{c}^1)}{P(\mathbf{r},\mathbf{w},\mathbf{c}^2)}, \quad (12)$$

where P_t is a probability distribution at the t-th episode. When observing the t-th episode, the prior distribution $P_{t-1}(\mathbf{h})$ is updated to the posterior distribution by calculating the likelihood and normalizing. Then, the posterior is used as the prior for learning from the next episode. Note that the $P(\mathbf{r},\mathbf{w},\mathbf{c}^2)$ is independent on the model because ($\mathbf{r}$, $\mathbf{w}$) and $\mathbf{c}^2$ are given from the observed data. Therefore, (12) is reformulated when the empirical distributions are used:

$$P_t(\mathbf{h},\mathbf{c}^1 \mid \mathbf{r},\mathbf{w},\mathbf{c}^2) \propto \prod_{d=1}^{D_t} \left\{ P(\mathbf{r}^{(d)},\mathbf{w}^{(d)} \mid \mathbf{h},\mathbf{c}^1,\mathbf{c}^2)P(\mathbf{c}^2 \mid \mathbf{c}^1)P(\mathbf{c}^1 \mid \mathbf{h})P_{t-1}(\mathbf{h}) \right\}, \quad (13)$$

The data generation term is divided into textual and visual features:

$$\log P(\mathbf{r}^{(d)},\mathbf{w}^{(d)} \mid \mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) = \sum_{n=1}^{N} \log P(r_n^{(d)} \mid \mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) + \sum_{m=1}^{M} \log P(w_m^{(d)} \mid \mathbf{c}^2,\mathbf{c}^1,\mathbf{h}). \quad (14)$$

Then the probability that the m-th element of the word vector is 1 is defined as follows:

$$P(w_m^{(d)} = 1 \mid \mathbf{c}^2,\mathbf{c}^1,\mathbf{h}) = \exp\left(s_m^{\mathbf{w}} - \sum_{i=1}^{|\mathbf{h}^{\mathbf{c}}|} \alpha_i \right), \ \mathbf{s}^{\mathbf{w}} = \sum_{i=1}^{|\mathbf{e}^{\mathbf{c}}|} \alpha_i e_i^{\mathbf{w}}, \quad (15)$$

where s_m is the m-th value of $\mathbf{s}$ and $\mathbf{e}^{\mathrm{c}}$ denotes the subpopulation of microcodes associated with $\mathbf{c}^1$. $e_i^{\mathbf{w}}$ denotes the textual and visual vectors of the i-th microcode. The probability of the image patches can be computed by the same way. The second term of (13) is related to predicting the characters from the mixtures of concrete concepts. It is defined to prefer more distinct concrete concepts for each character variable. The third term reflects the similarities of the subpopulation for each concrete concept node. The last term is determined from the used strategy of the graph MC. The weight of the microcodes is defined as a function of how frequently the words and patches of the microcode occur in the observed data. Whenever observing a new episode, the weight is updated:

$$\alpha_i^t = \lambda \alpha_i + (1-\lambda)\alpha_i^{t-1}, \quad (16)$$

where λ is a constant for moderating the ratio of the new observed episode and the previous episodes. In this study, we set λ to 0.9.

Vision-Language Conversion

The constructed visual-linguistic knowledge is used to convert scenes to text and vice versa, considering observed video stories. We view a vision-language conversion as a machine translation problem. Then, when source and target languages are substituted with scenes and subtitles, the vision-language translation is formulated:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{r},\theta) = \arg\max_{\mathbf{w}} P(\mathbf{r} \mid \mathbf{w},\theta)P(\mathbf{w},\theta), \quad (17)$$

where **w*** are the best subtitles generated from the constructed concept knowledge, and θ is a DCH model. **r*** can also be defined in the same way. Here **w*** is aligned to be a sentence by concatenating the words based on n-gram until a period appears and **r*** is synthesized to be a combination of the patches.

Experimental Results

Video Data Description and Parameter Setup

We use cartoon videos, called "*Pororo*", of 14 DVD titles with 183 episodes and 1,232 minutes of playing time. Pororo is a famous cartoon video appearing. By preprocessing, each scene is captured whenever a subtitle appears, transforming all the videos into the set of 16,000 scene-subtitle pairs. A scene image is represented by a bag of image patches extracted by maximally stable external regions (MSER), and each patch is defined as a feature vector by using SIFT and quantizing RGB pixel values.

We used a DCH model with two concept layers. A microcode consists of two image patches and a phrase with three consecutive words. The image patches are selected by UGMC and a phrase is selected with the maximum value of $P(v(x))$ of the words in the phrase. The initial number of $\mathbf{c}^1$-nodes starts at 10 and θ_{max} and θ_{min} are defined as follows:

$$\theta_{\max}^t = \begin{cases} \mu^t + \eta \cdot (\mu^t - \mu^{10}) \cdot \sigma^t, \ t > 10 \\ \mu^t + \eta \cdot \sigma^t, \ t \le 10 \end{cases}, \ \theta_{\min}^t = 0, \quad (18)$$

where μ^t and σ^t denote the mean and the standard deviation of the subgraph similarities after observing the t-th episode, and η is a constant for moderating the increasing speed of the $\mathbf{c}^1$ layer size. In this study, we set it to 0.75.

Concept Representation and Development

To demonstrate the evolution of concepts in DCH, we have examined how the characters, such as "*Pororo*", "*Eddy*", and "*Tongtong*", are differently described as the story unfolds. Figure 1 compares the descriptions after learning up to episode 13 (DVD 1) and 183 (DVD 14). Considering the fact that *Pororo* is a brother of *Crong*, *Tongtong* casts "*Kurikuri*" for magic, and *Eddy* is an engineer, the descriptive words for each character are suitable. We observe that the number of visual and linguistic nodes tends to increase. This is because the concepts continuously develop while observing the videos. The character concepts can be visualized as a multimodal

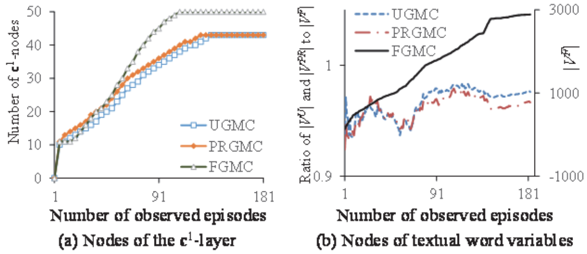


Figure 3: Changes of model complexity according to the learning strategies of the graph MC. In (b), V^U, V^{PR}, and V^F denote the vertex sets of the model constructed by UGMC, PRGMC, and FGMC.

concept map, which is used for knowledge representation as shown in Figure S1 in supplementary material. Specifically, we observed that the number of $\mathbf{c}^1$-nodes increases in early stages and then saturates (Figure 3). This indicates that new concrete concepts are learned rather earlier and, as time goes on, familiar concepts reappear. Figure 3(a) compares the complexity growth curves of DCH by three learning methods. FGMC is the most fast-growing strategy employing more $\mathbf{c}^1$-nodes because it tends to select diverse words and patches, as compared to UGMC and PRGMC. This is verified by Figure 3(b) which shows more vertices are included in the models constructed by FGMC. To see if DCH correctly learned the distinguishable concepts, we have analyzed the $\mathbf{c}^1$-nodes by PCA. Figure 4 shows that different characters are well discriminated by the learned microcodes (the first component ratio = 0.70 in (a)).

Vision-Language Translation Results

The constructed DCH was evaluated by using it for "story-aware" vision-language translation. Table 1 shows the performance of the sentence generation from the images. The test data consist of 183 images from randomly selecting one image per episode, and they are not used in training the models. The results are averaged over 10 experiments. The performances were estimated by how many words in the generated sentences and the original subtitles are matched. We examined how the different graph MC algorithms effect on the results. The precision of PRGMC increases faster in early videos but slower in late ones than that of FGMC. PRGMC is good at fast memorizing of main information but loses details. On the contrary, FGMC requires a more complex structure to memorize more information but shows higher accuracy. This is consistent with the results in Figure 3. In addition, the result shows that the introduction of concept layers improves the accuracy of the constructed knowledge. More examples of generated sentences and scene images are provided as supplementary material in Figures S2 and S3. It is interesting to note that the recall images are like mental imagery as demonstrated in movie recall in humans (Nishimoto et al. 2011). Overall, the results in Figure S2 and S3 demonstrate that the more episodes the DCH learned, the more diversity are generated in sentences and images. It should be noted that this is not for free; Observing more episodes requires heavier computational costs. The tradeoff should be made by the controlling the greediness of the graph MC algorithms as examined above.

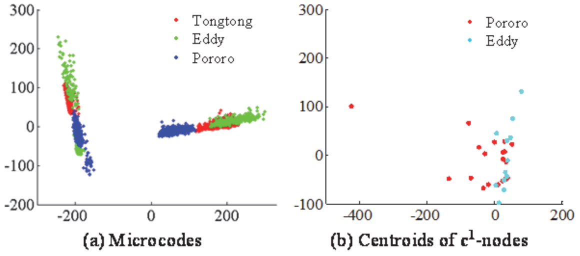


Figure 4: PCA plot of microcodes associated with the concrete concept nodes (**c1**-nodes) and their centroids of the models learned from 183 episodes by UGMC.

	Measure	PRGMC	UGMC	FGMC	SPC
Ep 1	Precision	**0.196**	0.077	0.167	0.195
	Recall	0.117	0.285	0.151	**0.302**
	F-score	0.146	0.122	0.158	**0.237**
Ep 1~9	Precision	0.225	0.221	**0.230**	0.175
	Recall	0.268	0.283	**0.303**	0.278
	F-score	0.245	0.248	**0.261**	0.215
Ep 1~18	Precision	0.240	0.247	**0.253**	0.239
	Recall	0.293	0.293	**0.378**	0.283
	F-score	0.264	0.268	**0.303**	0.259
Ep 1~36	Precision	0.267	0.251	**0.268**	0.242
	Recall	0.315	0.284	**0.376**	0.291
	F-score	0.289	0.266	**0.313**	0.264

Table 1: Performance of scene-to-sentence generation as the increase of the observed videos.

Concluding Remarks

We have presented a deep concept hierarchy (DCH) for automated knowledge construction by learning visual-linguistic concepts from cartoon videos. DCH represents mutually-grounded vision-language concepts by building multiple layers of hypergraph structures. Technically, the main difficulty is how to efficiently learn the complex hierarchical structures of DCH in online situations like videos. Our main idea was to use a Monte Carlo method. We have developed a graph MC method that essentially searches "stochastically" and "constructively" for a hierarchical hypergraph that best matches the empirical distribution of the observed data. Unlike other deep learning models, the DCH structure can be incrementally reorganized. This flexibility enables the model to handle concept drifts in stream data, as we have demonstrated in the experiments on a series of cartoon videos of 183 episodes.

We have analyzed and compared three strategies for the graph MC: uniform graph Monte Carlo (UGMC), poorer-richer graph Monte Carlo (PRGMC), and fair graph Monte Carlo (FGMC) depending on the probability of selecting vertices. The use of hierarchy improved the generalization performance while paying slight prices in computational cost. Among the variants of the Monte Carlo algorithms, we found that the PRGMC and the FGMC work better in earlier and later stages of video observation in the visual-language translation task. Overall, our experimental results demonstrate that DCH combined with the graph MC algorithms captures the mixed visual-linguistic concepts at multiple abstraction levels by sequentially estimating the probability distributions of visual and textual variables extracted from the video data. In future work, it would be interesting to see how the methods scale up on a much larger dataset with more complex story structures than the educational cartoon videos for children.

Acknowledgments

This work was supported by the National Research Foundation of Korea grant (NRF-2010-0017734). In addition, this work was supported in part by ICT R&D program of MSIP/IITP (10035348, 14-824-09-014, 10044009), the US Air Force Research Laboratory (AFOSR 124087), and Samsung Electronics Co., Ltd.

References

Abbott, J. T., Austerweil, J. L., and Griffiths, T. L. 2012. Constructing a Hypothesis Space from the Web for Large-scale Bayesian Word Learning. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society* (*Cogsci 2012*). 54-59.

Besag, J. 1974. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society*, *Series B*, 36(2):192-236.

Blei, D, M. and Jordan, M. 2003. Modeling Annotated Data. In *Proceedings of the 26th annual ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR 2003*). 127-134.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and L. Fei-Fei. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2009* (*CVPR 2009*). 248-255.

Fellbaum, C. 2010. WordNet, In R. Poli et al. (eds), *Theory and Applications of Ontology Applications*, pp. 231-243, Springer Science+Business Media.

Jia, Y., Abbott, J., Austerweil, J. L., Griffiths, T. L., and Darrell, T. 2013. Visual Concept Learning: Combining Machine Vision and Bayesian Generalization on Concept Hierarchies. *Advances in Neural Information Processing Systems 26* (*NIPS 2013*).

Kiefer, M. and Barsalou, L. W. 2013. Grounding the Human Conceptual System in Perception, Action, and Internal States, *Action Science: Foundation of an Emerging Discipline*, W. Prinz, M. Belsert, and A. Herwig (Ed.), MIT Press Scholarship Online.

Kiros, R., Salakutdinov, R., and Zemel, R. 2014. Multimodal Neural Language Models, In *Proceedings of the 31st International Conference on Machine Learning* (*ICML* 2014).

Lewis, M. and Frank, M. C. 2013. An Integrated Model of Concept Learning and Word-Concept Mapping. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (*Cogsci 2013*).

Meltzoff, A. N. 1990. Toward a Developmental Cognitive Science: The Implications of Cross-modal Matching and Imitation for Development of Representation and Memory in Infancy, *Annual New York Academy Science* 608:1-31.

Mikolov, T. Sutskever, I. Chen, K., Corrado, G., and Dean, J. Distributed Representation of Words and Phrases and Their Compositionality, In *Proceedings of Advances in Neural Information Processing Systems* (*NIPS 2013*). 3111-3119.

Muggleton, S. 2014. Alan Turing and the Development of Artificial Intelligence. *AI Communications*, 27:3-10.

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, L. 2011. Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies, *Current Biology* 21:1641-1646.

Ordonez, V., Kulkarni, G., and Berg, T. 2011. Im2text: Describing Images using 1 Million Captioned Photographs. In *Proceedings of Advances in Neural Information Processing Systems* (*NIPS* 2011), 1143–1151.

Quiroga, R. Q. 2012. Concept Cells: The Building Blocks of Declarative Memory Functions. *Nature Reviews Neuroscience* 13:587-597.

Socher, R., Ganjoo, M., Manning, C. D. and Ng, A. 2013. Zero-Shot Learning through Cross-modal Transfer. In *Proceedings of Advances in Neural Information Processing Systems* (*NIPS* 2013), 935–943.

Srivastava, N. and Salakutdinov, R. 2012. Multimodal Learning with Deep Boltzmann Machines. *Advances in Neural Information Processing Systems 2012* (*NIPS 2012*). 2222-2230.

Steyvers, M. and J. B. Tenenbaum. 2005. The Large-scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, 29:41-78.

Zhang, B.-T. 2013. Information-Theoretic Objective Functions for Lifelong Learning. *AAAI 2013 Spring Symposium on Lifelong Machine Learning*. 62-69.

Zhang, B.-T., Ha, J.-W., and Kang, M. 2012. Sparse Population Code Models of Word Learning in Concept Drift. In *Proceedings of the 34th Annual Conference of Cogitive Science Society* (*Cogsci* 2012). 1221-1226.

Zhao, G., Yuan, J., and Hua, G. 2013. Topical Video Object Discovery from Key Frames by Modeling Word Co-occurrence Prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* 2013 (*CVPR 2013*). 1602-1609.

Zhou, D., Huang, J., and Schoelkopf, B. 2007. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Proceedings of Advances in Neural Information Processing Systems 2006* (*NIPS 2006*). 1601-1608.