# Sparse Population Code Models of Word Learning in Concept Drift

**Byoung-Tak Zhang[1,2] (btzhang@bi.snu.ac.kr)**
**Jung-Woo Ha[1] (jwha@bi.snu.ac.kr)**
**Myunggu Kang[1] (mgkang@bi.snu.ac.kr)**
[1]School of Computer Science and Engineering
[2]Cognitive Science and Brain Science Programs
Seoul National University, Seoul 151-744, Korea

**Abstract**

Computational modeling has served a powerful tool for studying cross-situational word learning. Previous research has focused on convergence behaviors in a static environment, ignoring dynamic cognitive aspects of concept change. Here we investigate concept drift in word learning in story-telling situations. Informed by findings in cognitive neuroscience, we hypothesize that a large ensemble of sparse codes flexibly represents and robustly traces drifting concepts. We experimentally test the population coding hypothesis on children's cartoon videos. Our results show that learning the meanings of words over time is hard, especially when the concept evolves slowly, but the sparse population coding can handle the concept drift problem effectively while hypothesis elimination and simplistic parametric models have difficulty.

**Keywords:** Cross-situational word learning; statistical language learning; concept drift; meaning change; population coding.

## Introduction

Children learn the meaning of words rapidly and robustly across multiple situations (Smith & Yu, 2008). Computational modeling has served a powerful tool for precise investigation of the hypothesized mechanisms of word learning. Many computational models of word learning have been used to simulate and account for the observed patterns such as reference disambiguation, blocking, and long-term memory (Frank *et al.*, 2009, Kachergis *et al.*, 2010; Vlach & Sandhofer, 2010).

Existing computational models for word learning can be broadly divided into hypothesis elimination and associative learning (Fazly *et al*., 2010). In the hypothesis elimination approach the learning process consists of eliminating incorrect hypotheses about word meaning, on the basis of a combination of a priori knowledge and observations of how words are used to refer to aspects of experience, until the learner converges on a single consistent hypothesis. For instance, Siskind (1996) presented an efficient algorithm for keeping track of just the necessary and possible components of word-meaning hypotheses consistent with a set of examples. A weakness of this approach is that some logically possible hypotheses may be ruled out a priori or the concepts cannot be recovered once they are eliminated.

Another approach to computational modeling of word learning is associative learning. Yu (2005), for example, studied a word-object association model in a unified framework of lexical and category learning. This model demonstrated the emergence of patterns observed in early word learning. Xu and Tenenbaum (2007) proposed a probabilistic model of word learning. The Bayesian account aims to explain inductive learning at the level of computational theory rather than to describe psychological processes involved. Fazly *et al.* (2010) uses a probabilistic framework to propose an incremental associative model that deals with referential uncertainty. The proposed model is demonstrated to converge over time on the most likely meaning of the word in CHILDES data sets. However, this model does not incorporate alignment ambiguity and it is not clear how the model behaves if the concept drifts in the course of learning.

Concept drift is a fundamentally important phenomenon in language acquisition. It means that the statistical properties of the target concept, which the learner is trying to learn, change over time (Widmer & Kubat, 1996). For example, a child might think that all birds can fly until he/she observes an ostrich, at which time the child revises the concept of bird. This causes problems because the learning process needs some mechanisms to unlearn or revise the learned concepts. Simple hypothesis elimination cannot account for this since it lacks a mechanism for recovering the eliminated concepts. Both the associative learning and its probabilistic versions have difficulties since they strive to model global patterns, not modeling local patterns that might be necessary at a later stage.

Here we propose a computational model of word learning that deals with concept drift under alignment ambiguity and referential uncertainty. The model borrows ideas from neuroscience and uses a population coding (Pouget *et al.*, 2000; Ma *et al.*, 2006). We propose a sparse population-code network in which meanings of the words are represented as a large collection of sparse microcodes. Since each microcode is sparse, it describes a general concept. There are many of the microcodes and, thus statistically, only a few parts of them are updated on a single observation, maintaining stability by the remaining microcodes in the population. We test this population coding hypothesis on naturalistic children's cartoon video data. To make the experiments more realistic, we use state-of-the-art image processing techniques to represent the scene as a bag of image patches. This is contrasted with the previous studies of cross-situational word learning in which the scene representation adopts hand-coded semantic features. Our experimental results show that learning the meaning of words over time is hard, especially when the concept is drifting slowly. We demonstrate that the sparse population coding can handle the concept drift problem effectively

while simplistic parametric models have difficulty in dealing with the problem.

## Materials and Experimental Setup

### Video Data Sets

We used a series of children's cartoon videos, *Maisy*, consisting of 6 episodes. Each episode plays for 48 to 105 minutes and the total play time is 475 minutes. From this video set, we prepared a total of 972 utterance-scene pairs as described in the following subsections. Cartoon videos provide naturalistic story-telling situations that children face in language acquisition (Zhang & Kang, 2011). An additional advantage of cartoons is that its image processing is relatively easy, allowing for automated generation of a large data set to study the long-term learning behavior in situated word learning.

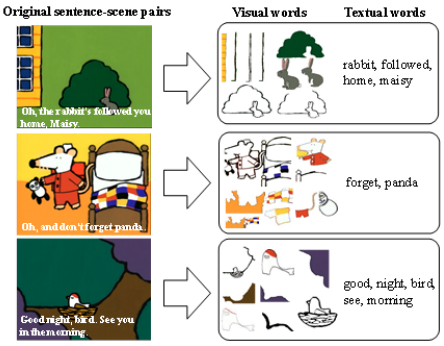



Figure 1: Examples of utterance-scene representation

### Utterance-Scene Representation

The material for cross-situational learning consists of utterance-scene pairs, where each pair is represented as a vector of the form

$$\mathbf{x}(t) = (\mathbf{w}(t), \mathbf{v}(t)) = (w_1, ..., w_{|\mathbf{w}(t)|}, v_1, ..., v_{|\mathbf{v}(t)|}) .$$

Here, $|\mathbf{w}(t)|$ and $|\mathbf{v}(t)|$ are the number (vocabulary size) of textual words and visual words in the $t$-th example, respectively. Figure 1 shows the examples of utterance-scene pairs extracted from the original videos. The following subsection describes how the textual words were processed.

### Language Processing

We collected all utterances in the text captions of the video set, which amounts to approximately 2,800. Removing simple utterances such as 'Hi' gives a total of 972 sentences. We determined the vocabulary for textual words by computing the standard TF-IDF (term-frequency and inverse-document-frequency) values. TF-IDF gives higher weights to the terms that frequently occur and are uncommon between episodes. This results in 1,049 words. We chose the top 448 textual words which defines the utterance vocabulary. The sound modality was not used in the experiments.

### Image Processing

We extracted image frames from the video, one frame for each of the 972 sentences extracted by language processing. Out of a stream of image frames played for the duration of speech of an utterance, we chose the image frame corresponding to the start of the utterance. This results in an image corpus of 972 scenes. Each scene was described by a subset of 7,520 image patches (i.e., visual words), each composed of the SIFT (scale-invariant feature transform) features and the color histogram extracted as follows. To define the visual words, we first used the MSER (maximally stable extremal region) feature extractor to segment and extract salient and informative regions from the images. SIFT was then used to find salient features in the extracted regions. The resulting features are grouped by K-means clustering to remove redundancy.

### Experimental Paradigm

Given the set of learning examples $D_N = \{(\mathbf{w}(t), \mathbf{v}(t)) \mid t = 1, ..., N\}$, the goal of the learner is to form the concepts in the training set by finding the relationships between the words and the visual words (i.e. image patches). Learning proceeds incrementally, i.e. the examples are presented in sequence. Each time an example is presented the learner updates its model before the next example comes in.

Figure 2(b) shows the paradigm we adopt in this study. As indicated by the connections between textual words and those between visual words, we consider the fully interconnected relationship between different words and visual words. Note that this paradigm is contrasted with the standard paradigm shown in Figure 2(a), where the learner is to learn the relationship between the words and the referents or meanings, but do not attempt to learn the relationship among the words or among the referents.

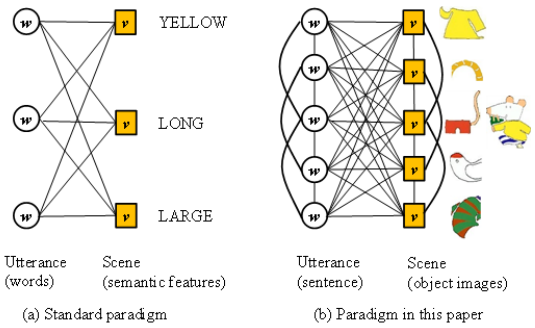



Figure 2: Experimental paradigms in comparison

## The Model

### Concept Representation

Meaning of words can be defined as a set of contexts in which word occurs in running text (Burgess & Lund, 1998) or represented in network connectivity revealed by statistical analysis of a text corpus (Steyvers & Tenenbaum,

2005). The textual domain can be extended to include the visual domain by taking into account the full contexts in which the word and images (visual words) co-occur in scenes (Zhang, 2008). Figure 3 illustrates this type of concept representation we adopt in this work. Here, the concept of MOUSE, for example, is defined as a collection of words (*w*-nodes), i.e. {yellow, run, dark, tall}, and a collection of visual words or visual patches (*v*-nodes) linked to the 'mouse'-node in the figure. Thus, we consider the learner to acquire the visually-grounded linguistic concepts or the joint vision-language concepts, similar to the perceptual symbol systems *a la* Barsalou (1999).

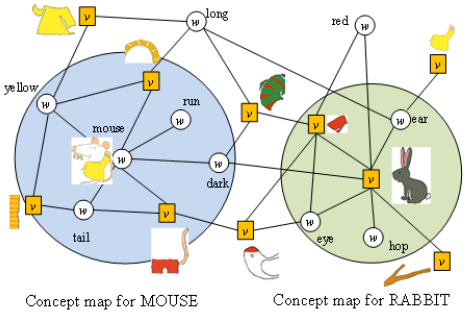



Figure 3: The concept of a word is defined by other words *w* as well as visual words *v*. In this representation the concepts are defined as a relationship among the primitives (words and visual words).

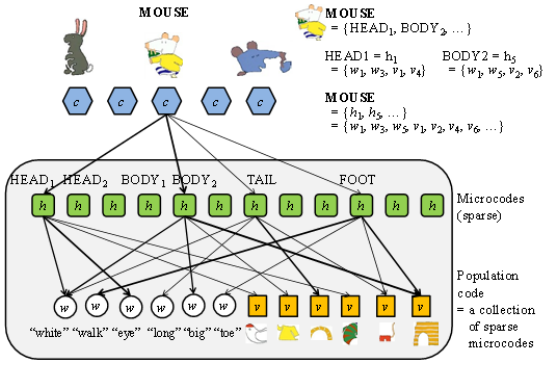



Figure 4: Sparse population coding scheme

## Sparse Population Coding

We represent the joint vision-language concept using sparse population codes. Figure 4 shows the basic units of the coding scheme, i.e. microcodes. Each microcode represents a prototype, exemplar, or common pattern for a set of similar examples. For instance, a microcode $h$ = {'white', 'eye', $v_1$, $v_4$} represents a class of objects (or concept HEAD1 as indicated in the figure) that have white eyes and image features of $v_1$ and $v_4$, where $v_1$ and $v_4$ are image patches. The textual words, 'white' and 'eye', and the visual words, $v_1$ and $v_4$, are instances of the textual and visual word vocabulary, respectively. Since the number of words or visual words chosen to define the specific microcode is small compared to their vocabulary size, this is a sparse coding scheme. Typically we use a large number of microcodes to describe complex concepts.

The population of sparse microcodes can be considered as a three-layer network as shown in Figure 4. The first (bottom) layer consists of the *w*-nodes for words (e.g. "white") and the *v*-nodes for visual words (image patches). The second (middle) layer represents the *h*-nodes for microcodes or micro-concepts such as HEAD1. A formal concept is represented as an ensemble of micro-concepts (or microcodes), as indicated by *c*-nodes at the third (top) layer of the network. This network can be learned from the data. Before describing the learning procedure we see the statistical background underlying this representation.

## Finite Mixture Model Formulation

Formally, a large collection of microcodes represents the empirical distribution of the concepts in the form of a finite mixture model (McLachlan & Peel, 2000). To see this, we suppose that the density of data $\mathbf{x} = (\mathbf{w},\mathbf{v})$ can be written in the form:

$$P(\mathbf{x} \mid \theta) = \sum_{j=1}^{M} \alpha_j f_j\left(\mathbf{x} \mid h_j\right) \qquad (1)$$

where $f_j(\mathbf{x}|h_j)$ are densities and $\alpha_j$ are nonnegative quantities that sum to one:

$$0 \le \alpha_j \le 1 \quad (j = 1,...,M) \quad \text{and} \quad \sum_{j=1}^{M} \alpha_j = 1.$$

Equation (1) is called $M$-component finite mixture density. Roughly, the configuration of the microcode defines the shape of the mixture component $f_j(\mathbf{x}|h_j)$ and the weight associated with the microcode defines the mixing weight $\alpha_j$. We denote the complete collection of all distinct parameters occurring in the mixture model by $\theta = (\boldsymbol{\alpha},\mathbf{h})$, where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_M)$ and $\mathbf{h} = (h_1,\ldots,h_M)$. We note that by designing the microcodes $h_j$ appropriately to be the parameters of the component density $f_j(\mathbf{x}|h_j)$, the mixture density can be represented by the sparse population code.

In other words, if the microcode has an associated component density $f_j(\mathbf{x}|h_j)$, the distribution of the data set $D_N = \{\mathbf{x}(1),...,\mathbf{x}(N)\}$ can be represented by the population code:

$$P(\mathbf{w}(1),\mathbf{v}(1),...,\mathbf{w}(N),\mathbf{v}(N) \mid \theta)$$
$$= P(\mathbf{x}(1),...,\mathbf{x}(N) \mid \theta) = \prod_{t=1}^{N} \sum_{j=1}^{M} \alpha_j f_j(\mathbf{x}(t) \mid h_j) \qquad (2)$$

which is a sum of $M^N$ products of component densities. Each term in the summation is interpreted as the probability of obtaining a given one of the MN possible divisions of the observations among the groups.

## Learning Algorithm

Learning proceeds incrementally by observing each utterance-scene pair in sequence. On each observation of an

example $(\mathbf{w}, \mathbf{v})$ the learner predicts and updates the meanings or concepts $\theta$. This is an inductive process and can be formally described as Bayesian inference:

$$P_t(\theta \mid \mathbf{w}, \mathbf{v}) = \frac{P(\mathbf{w}, \mathbf{v} \mid \theta) P_{t-1}(\theta)}{P(\mathbf{w}, \mathbf{v})} \quad (3)$$

At each time step $t$, the prior distribution $P_{t-1}(\theta)$ of the hypothesis $\theta$ is updated to the posterior distribution $P_t(\theta|\mathbf{w},\mathbf{v})$ of the hypotheses by computing the likelihood function $P(\mathbf{w}, \mathbf{v} \mid \theta)$ and normalizing by

$$P(\mathbf{w}, \mathbf{v}) = \sum_{\theta'} P(\mathbf{w}, \mathbf{v} \mid \theta') P_{t-1}(\theta') \quad (4)$$

to make $P_t(\theta|\mathbf{w},\mathbf{v})$ back to a probability distribution. The posterior is then used as the prior for the next time step. Making the data set explicit, we can rewrite (3) in a recursive form:

$$P_t(\theta \mid \mathbf{w}(t), \mathbf{v}(t), \mathbf{w}(1:t-1), \mathbf{v}(1:t-1)) = \frac{P(\mathbf{w}(t), \mathbf{v}(t) \mid \theta) P_{t-1}(\theta \mid \mathbf{w}(1:t-1), \mathbf{v}(1:t-1))}{P(\mathbf{w}(t), \mathbf{v}(t) \mid \mathbf{w}(1:t-1), \mathbf{v}(1:t-1))}, \quad (5)$$

where $\mathbf{w}(t)$ and $\mathbf{w}(1{:}t\text{-}1)$ denote the word vector at time step $t$ and the sequence of word vectors from time step 1 to $t$-1, respectively. Expectation-maximization (EM) style algorithms are usually used to solve the estimation problem (McLachlan & Peel, 2000). In the following we describe the method we implemented as a sparse population coding network. Recall that $\theta = (\boldsymbol{\alpha}, \mathbf{h})$, i.e. the concepts are represented as a collection of microcodes $\mathbf{h}$ with weights $\boldsymbol{\alpha}$ in the network. The population code is a mechanistic representation for psychological processes since it describes the memory encoding and decoding mechanisms more explicitly than simplistic parameter tuning models.

We first describe the learning algorithm in pseudocode and then explain it.

1 $H(0) \leftarrow \{\}$, $V_T \leftarrow \{\}$, $V_I \leftarrow \{\}$
2 $t \leftarrow 1$ ; prior $P_{t-1}(\theta)$
3 Perceive $\mathbf{x}(t) = (\mathbf{w}(t), \mathbf{v}(t))$
4 $E = \{h_1, \ldots, h_m\} \leftarrow$ Sample($\mathbf{x}(t)$) ; microcodes
5 $V_T \leftarrow V_T$ + {new *w's*}, $V_I \leftarrow V_I$ + {new *v's*}
6 $H' \leftarrow H + E$ ; *accommodation*
7 Repeat
8 $H'' \leftarrow$ Predict($H'$) ; sampling prior $P_{t-1}(\theta)$
9 $\mathbf{x}' \leftarrow$ Generate($H''$) ; likelihood $P(\mathbf{x} \mid \theta)$
10 $H''' \leftarrow$ Correct($H''$, $\mathbf{x}'$, $\mathbf{x}(t)$) ; *assimilation* (resampling)
11 $H' \leftarrow H'''$
12 Until *reconstruction_satisfactory*($\mathbf{x}(t)$)
13 $H(t) \leftarrow H'$ ; posterior $P_t(\theta \mid \mathbf{x}(t))$
14 $t \leftarrow t+1$
15 Go to 3

Given an utterance-scene instance (line 3), a subset of words and a subset of visual words are selected to build a microcode (line 4). For each utterance-scene pair, a number $m$ of microcodes are generated randomly and repeatedly (line 4). Duplications are permitted and, in fact, the number of duplications represents the strength of the code (we will use this later on in decoding the referents or meanings of the words). The set $E$ of new microcodes is then added to the existing set $H$ of microcodes (line 6). This step is equivalent to accommodating new memory elements. Then the model is trained to tune or assimilate the incoming concepts into the existing concepts (lines 7-12). First, a collection $H''$ of the microcodes is sampled to be used to generate an example $\mathbf{x}'$. The generated example is then compared to the training example. The difference is used to correct the model $H$ or the population code. This results in the update of the posterior distribution (line 13).

The algorithm consists of basically three steps: i) sampling new microcodes (line 4), ii) merging them with the old (existing) population of microcodes (line 6), and iii) resampling of the whole microcode population (line 10) to correct the conflicts and interferences. To correct predictive errors in an unsupervised way, the algorithm test-generates the samples from the current model (lines 8 and 9) and compare the resulting data with the perceived data (line 10).

### Connection to Probabilistic Models of Cognition and Monte Carlo

Recall that the population of sparse codes approximates the probability distribution of the examples if the population size is big. Recall also that the learning algorithm is implemented by repeatedly sampling the sparse codes (microconcepts or hypotheses) like a Monte Carlo simulation does. In terms of Bayesian inference the learning algorithm updates the distribution of the concepts from prior to posterior distribution by Monte Carlo simulation. Shi *et al.* (2010) suggested that exemplar models are a successful class of psychological process models that can be used to perform a sophisticated form of Monte Carlo approximation. The similarity of the sparse coding representation with the exemplar model suggests that our sparse population code model offers a concrete process model of Bayesian cognition.

## Simulations and Results

### Parameter Setting for Experiments

Experiments were performed using the following parameter settings. Given a new observation, 10 new sparse codes were sampled and added to the population. Each microcode consists of three textual words and one visual word. 5 iterations of error correcting steps were executed to tune the whole population code to the new observation. To see the effects of memory capacity we experimented with two sizes of populations: $|H|$ = 100, 500. When the population size exceeds the memory capacity, we replace 10 microcodes with the lowest weight values by 10 new microcodes. We define two scores for measuring the similarity between visual and textual concepts as follows:

$$S(w,v) = \frac{1}{\sum_{h \in H_w} \alpha(h)} \cdot \left(|H_w|\right)^{\frac{1}{2}} \cdot \sum_{h \in H_v} \alpha(h) \text{ and } S(w) = \sum_{h \in H_w} \alpha(h)$$

where $\alpha(h)$ is the weight of microcode $h$, and $H_w$ and $H_v$ are the subsets of $H$ consisting of the microcodes with textual word $w$ and visual patch $v$, respectively.

## Vocabulary Growth

Figure 5 shows the growth of visual and textual vocabularies as learning proceeds. When the memory size is unlimited (left), the size of both visual words and textual words increases continuously (linearly). When the maximum memory capacity is set to be limited to 500 (right), the size of visual words increases first and then decreases while the number of textual words grow continuously but in two stages of fast growth and then slow growth. The difference in vocabulary growth pattern seems in part due to the difference in vocabulary size of visual and textual words, i.e. in this experimental setting, 7520 visual words and 448 textual words were used for candidates.

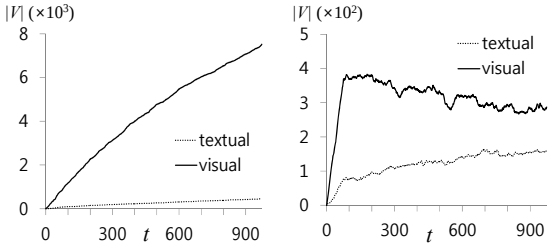



Figure 5: Growth of vocabulary. (left) unlimited memory size. (right) limited memory size.

## Word Learning in Concept Drift

Figure 6 shows the trace of concept memory for the 4 separate focus objects which appear in all episodes.

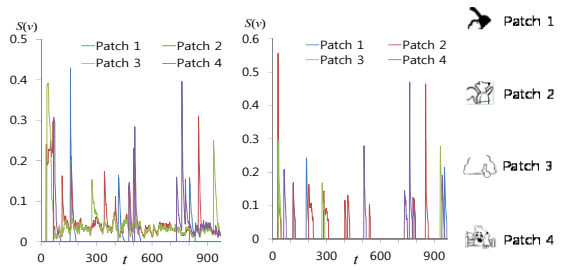



Figure 6: Emergence, extinction, and re-emergence of concepts in drift. (left) larger memory capacity ($|H|$ = 500). (right) smaller memory capacity ($|H|$ = 100).

The results show the emergence, extinction, and re-emergence of different visual concepts as the video runs. If the memory size is relatively big (500 in this case), the concepts do not extinct totally and remain in the backend to re-emerge when new similar observations are made. In contrast, when the memory size is small (100 in this case), the concepts disappear entirely from the memory, suggesting the difficulty of the problem, especially if the memory capacity is small. However, this problem can be solved by dynamically varying the population size to balance exploration and exploitation. In contrast to this sparse population coding approach, a localist, eliminative method would have a fundamental difficulty in recovering once-eliminated concepts due to its lack of associative connections between concepts.

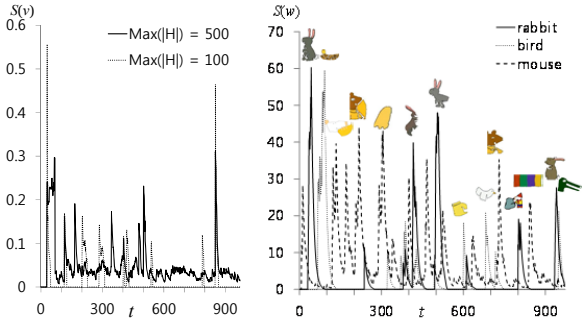
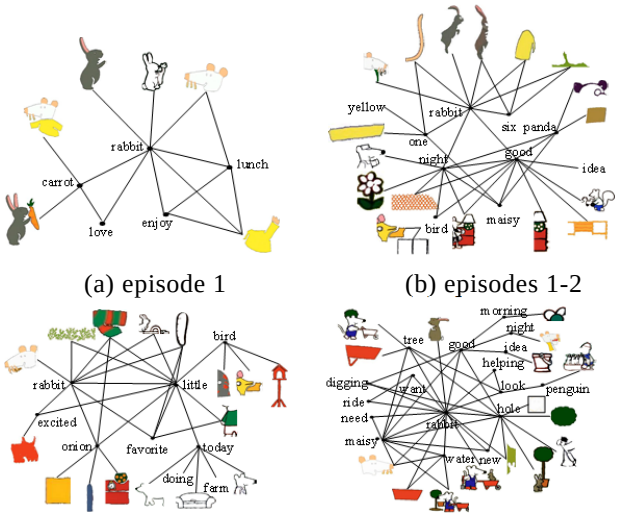



Figure 7: (left) Emergence patterns of concepts for different memory size. Plotted are the weight values for the specific visual concept shown. (right) Emergence of different visual concepts for given three textual concept (rabbit, bird, and mouse).

Figure 7 shows the change of concepts in the course of learning. (left) shows the change of weight distribution for the specific visual concept (patch 2 in Figure 6) shown. (right) is the reverse, i.e. the query is given by a textual word (rabbit) and the graph shows how the corresponding image concept changes as learning proceeds. It can be observed that, since the concept of rabbit drifts, different types of rabbit images and, sometimes very different (and wrong) images, are retrieved by the same word.

## Concept Generalization and Specialization

Figure 8 shows the joint vision-language concept maps around the 'rabbit' as they evolve over the 6 episodes. The maps (a)-(d) are the snapshots after watching 1, 2, 4, 6 videos, respectively. Note that the map contains visual words as well as textual words. We observe that the connectivity of the visual-linguistic map grows as more episodes are learned. Careful examination of the map shows the role of visual words or concepts for the specialization and generalization of the textual concepts and vice versa. For example, in Figure 8(a) we observe that a visual word connects the three textual words of 'enjoy', 'lunch', and 'rabbit' together. This adds an additional, visually-grounded, connection (association) between the words 'enjoy' and 'lunch'. We also observe that the word 'rabbit' is connected to multiple images, again grounding and refining the meaning of the textual word. Formally, the former is the one-image to many-words relationship and the latter is the many-images to one-word relationship. This again shows the effect of visual generalization of the textual words and that of visual specialization, respectively, which cannot be observed in language-only concept maps (Zhang & Kang, 2011).




(a) episode 1 (b) episodes 1-2

(c) episodes 1-4 (d) episodes 1-6

Figure 8: Evolution of the vision-language concept map for 'rabbit'.

## Discussion

We have presented a sparse population code model of cross-situated word learning in concept drift. The sparse population coding was utilized to flexibly represent and learn the meanings of words over time. We examined the concept drift in word learning using the story-telling situations in cartoon videos. The experimental results demonstrate that the model is effective in learning the dynamically changing meanings of the words.

We adopted a distributed, relational representation of word meaning which is naturally realized as a population of sparse codes. The learning process constructs visually-grounded linguistic knowledge structure from a series of cross-situational language experience. This situated conceptualization process (Barsalou, 1999) is known to build a foundational mechanism in language learning (Zwaan & Kaschak, 2008). We analyzed the "evolution" of joint vision-language maps and compared them to the language-only concept maps. We found that the visual modality adds additional semantics to the linguistic concepts as well as generalizing and/or specializing the linguistic terms.

There are several directions of future research that can extend the current work. One is to define the vocabulary incrementally. The current experiments have used a set of words and visual words which are defined at the outset. Children learn the new words on the fly. A more natural approach would employ a component that evaluates the novelty and decide the introduction of new terms. Another direction involves extending the current population-code network model by introducing another layer of latent variables. This layer can be learned to build more abstract concept categories using, for example, non-parametric Bayesian methods.

## Acknowledgments

This work was supported by the National Research Foundation (NRF) grants (0421-20110032-Videome, 2010-0018950-BrainNet), the IT R&D Program of KEIT (10035348-mLife), and the BK21-IT Program.

## References

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577-609.

Burgess, C. & Lund, K. (1998). The dynamics of meaning in memory. In Dietrich & Markman (Eds.), *Cognitive Dynamics: Conceptual Change in Humans and Machines.*

Fazly, A., Alishahi, A., & Stevenson, S. (2010). A probabilistic computational model of cross-situational word learning. *Cognitive Science*, 34, 1017-1063.

Frank, M.C., Goodman, N.D., & Tenenbaum, J.B. (2009). Using speakers' referential intentions to model early cross-situational word learning. *Psychological Science,* 20, 579-585.

Kachergis, G., Yu, C., & Shiffrin, R. M. (2010). Adaptive constraints and inference in cross-situational word learning. *Proceedings of 2010 Annual Conference of the Cognitive Science Society* (pp. 2464-2469).

Ma, W., Beck, J., Latham, P., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9 (11), 1432-1438.

McLachlan, G.J., & Peel, D. (2000). *Finite mixture models.* Wiley.

Pouget A, Dayan P, & Zemel, R. (2000). Information processing with population codes. *Nature Reviews Neuroscience*, 1(2), 125-32.

Shi, L., Griffiths, T. L., Feldman, N. H., & Sanborn, A. N. (2010). Exemplar models as a mechanism for performing Bayesian inference. *Psychonomic Bulletin & Review*, 17(4), 443-464.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(12), 1–38.

Smith, L., & Yu, C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106(3), 1558–1568.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: statistical analyses and a model of semantic growth. *Cognitive Science, 29*, 41-78

Vlach, H. A., & Sandhofer, C. M. (2010). Desirable difficulties in cross-situational word learning. *Proceedings of 2010 Annual Conference of the Cognitive Science Society* (pp. 2470-2475).

Widmer, G. & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts**,** *Machine Learning*, 23, 69-101.

Xu, F., & Tenenbaum, J. B. (2007). Word learning as Bayesian inference. *Psychological Review*, 114(2), 245-271.

Yu, C. (2005). The emergence of links between lexical acquisition and object categorization: A computational study. *Connection Science*, 17(3–4), 381–397.

Zhang, B.-T. (2008). Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory, *IEEE Computational Intelligence Magazine*, 3(3), 49-63.

Zhang, B.-T., & Kang, M.-G. (2011). Bayesian mixture modeling of joint vision-language concepts from videos, *NIPS-2011 Workshop on Integrating Language and Vision.*

Zwaan, R.A. & Kaschak, M.P. (2008) Language in the brain, body, and world. Chap.19. *Cambridge handbook of situated cognition*.