Weakly-supervised word learning is improved by an active online algorithm

Heikki Rasilo[1,2], *Okko Räsänen*[1]

[1] Aalto University, Dept. Signal Processing and Acoustics, Otakaari 5 A, 02150 Espoo, Finland
[2] Vrije Universiteit Brussel, Artificial Intelligence Laboratory, Pleinlaan 2, 1050 Elsene, Belgium
heikki.rasilo@aalto.fi, okko.rasanen@aalto.fi

Abstract

When infants learn words, they do not generally occur in isolation but as parts of continuous utterances and with several possible related meanings. Details of the efficient learning techniques of infants remain unknown. In this paper we introduce a dynamic concept matrix (DCM) algorithm that learns acoustic models for a set of keywords from given pairs of continuous utterances and related keyword labels. DCM is an incremental, active online algorithm. Specifically, each training utterance is first recognized with the current word models, and the recognition result is used to guide training further. In low-noise conditions DCM shows significant improvement in convergence rate and final recognition scores to an earlier passive CM model on TIDIGITS and CAREGIVER UK Y2 datasets. The results suggest that in ambiguous learning situations it may be beneficial for the learner to observe the learning situation, make hard decisions if some known words/objects were recognized and update the models based on the decisions.

Index Terms: weakly-supervised word learning, incremental model, online algorithm, cross-situational learning

1. Introduction

Infants learn meanings of words in their native language efficiently without explicit supervised teaching. The details of human word-learning mechanisms still remain unknown – learning is complicated by vast amounts of variability in continuous speech utterances as well as lack of known robust cues, such as pauses, marking word boundaries. There is evidence that infants learn basic word forms based on statistical dependencies between consecutive speech sounds [1], but automatic blind speech segmentation algorithms have had only limited success in word learning mainly due to speech variability, and tend to learn frequently occurring patterns that can also be parts of words or sequences of words (e.g. [2]).

Since infants constantly interact with their environment and surrounding people, heard speech is often related to surrounding objects or actions, thus providing information about their possible *meanings*. When a meaning co-occurs with a certain acoustic pattern in several learning situations, (e.g. acoustic sequences “lookatdaddy” and “daddyishere” occur with the visual percept of father), the infant can infer that the only consistently occurring acoustic pattern could refer to the seen object (“daddy” – father). This learning strategy is often called cross-situational learning (XLS, [3]) and is a widely studied phenomenon when learning word-meaning mappings (e.g. [4]) for example. However, most XSL studies focus on learning meanings for words that are already segmented out of continuous speech, thus avoiding the problems caused by acoustic variability faced by real infants.

In this paper we improve an existing algorithm that learns word models from real acoustic speech streams using the XSL paradigm. Our goal is not to compete with supervised state-of-the-art word or speech recognition algorithms. In engineering terms our algorithm can be considered *a weakly supervised word discovery algorithm* – an algorithm that learns from its surroundings independently when provided with continuous speech streams and unordered and possibly noisy sets of related meanings at the time-scale of one or more utterances.

1.1. XSL on acoustic speech data

A few studies have contributed to performing XSL on recorded acoustic speech. Acoustic models for words are trained using a speech database, where each utterance comes with an unordered set of labels that correspond to some keywords in the utterance. After training, the algorithm should have converged to acoustic models for every keyword, allowing accurate recognition of the keywords in a novel test set.

Non-negative matrix factorization (NMF) has been used in batch mode (all training data is available for the learner at once) [5] and in incremental mode (associations form gradually when new learning scenarios occur) (Adaptive NMF, [6], and its active adaptation in [7]) successfully in learning to derive correct meanings for words in novel test utterances. Also adaptive Bayesian probabilistic latent semantic analysis [8] has succeeded in this task. However, the adaptive algorithms have been evaluated with CAREGIVER Y1 data [9] in learning scenarios where each training (testing) utterance has only one meaning to be learned (recognized). These scenarios lack the ambiguity present in normal XSL experiments where multiple candidate meanings per word are present. Word recognition using NMF also generally leads to a list of hypothesized words present in a complete utterance – speech is not segmented and the ordering or locations of the words in the utterances are thus not solved ([10], [5]). However, a sliding window decoder has been proposed to partially solve the ordering problem [5].

Räsänen and Laine [11] have proposed an incremental weakly-supervised pattern discovery algorithm (*concept matrix*, CM), that learns words and their segmentation from weakly-labeled speech data incrementally. The algorithm is given speech utterances and an unordered set of keyword labels (= concepts) per utterance. Accumulating information from several utterances, the algorithm learns acoustic models for words and is able to recognize words accurately from novel test utterances. The algorithm has been tested with the TIDIGITS (from here on TI) corpus and the CAREGIVER Y2 corpus (Y2), showing that the CM can learn robust acoustic models for words from ambiguous utterances and without ever being shown the words in isolation.

1.2. Active vs. passive online algorithms

The basic CM is an incremental, online algorithm in the sense that training data is processed utterance by utterance and word models are updated at every trial. However, the training algorithm treats all training utterances equally, and thus for example the ordering of the training utterances does not affect performance. The basic CM algorithm can thus be considered a passive online algorithm. In human XSL, ordering of the training data affects learning results (e.g. [12]) and humans use information acquired earlier to deduce further word meaning mappings (for example in *mutual exclusivity*, see e.g. [13]). Human learning thus is active – learning situations are actively analyzed and learning depends on earlier knowledge.

Previous research has shown that in some tasks active online processing of training data can lead to better performance than processing in batch mode. Online training of HMMs, so that the recognizer's parameters are re-estimated based on the error between the decoded training utterances and the ground truth, has been shown to improve convergence rate and accuracy on a supervised phoneme recognition task [14]. Online processing and memory constraints have also improved the performance of Bayesian word segmentation based on pre-syllabified speech data [15]. Similarly, an online implementation of EM algorithm has led to improvement in part-of-speech tagging, document classification, English word segmentation from phoneme sequences and word alignment [16]. Fazly et al. [17] have used an active training method when learning word-meaning mappings from readily segmented speech. They update word-meaning pairings based on the alignment probabilities obtained in previous training trials and show efficient word learning and robustness towards noise.

In this paper we propose an active version of the original CM algorithm, and call it Dynamic Concept Matrix (DCM). DCM uses information collected from previous utterances to infer locations of keywords in an utterance, and weight the training of corresponding word models to these locations. Each training utterance is thus recognized with the current models, and the recognition result affects the training of the utterance. The algorithm shows improved rate of learning as well as word recognition performance with real acoustic speech in Y2 and TI corpuses in low noise conditions.

2. Method

2.1. The concept matrix technique

The CM algorithm constructs acoustic models for given concepts (= keyword labels in this work) by approximating high-order Markov structure of speech as a mixture of bi-gram statistics at different temporal lags, and then providing maximum-likelihood estimates for each referential context given the currently observed sequence of acoustic observation [11]. For example an acoustic utterance "Smiling **daddy has** the **fish**", where keywords are bold and the rest are *carrier words* belonging to a *carrier sentence*, would be first transformed into a sequence of integers by vector quantization (VQ) of spectral slices. Then initially empty concept matrices for the keywords "daddy", "to have" and "fish" would be updated by counting frequencies of transitions between these integers.

More formally, every keyword label *c,* from a set of possible labels **C** has a three dimensional frequency matrix $\mathbf{F}_c$ of size $N_U \times N_U \times L$, where N_U is the number of possible VQ-indices and L is the total number of lags used from a pre-defined set $\boldsymbol{l} = \{l_1, l_2, \ldots, l_L\}$. When a sequence of VQ-indices $X = [u_1, \ldots, u_{t-1}, u_t, u_{t+1}, \ldots, u_{t+m}]$, and a set of related labels $\boldsymbol{c} \subseteq \mathbf{C}$ are given, the frequency matrices for every given label c at every time instant t and every lag l are updated as

$$\mathbf{F}_c(u_t, u_{t+l}, l) = \mathbf{F}_c(u_t, u_{t+l}, l) + a \qquad (1)$$

Where $a = 1$ always in the basic CM algorithm and the variables inside the brackets refer to the matrix elements. After training with all sequences and labels in the training set, the matrices are normalized first to represent transition probabilities P_T between elements (e.g. normalization over the second dimension of the matrices)

$$\mathbf{P}_T\left(u_j \middle| u_i, l, c\right) = \mathbf{F}_c\left(u_i, u_j, l\right) / \sum_{j=1}^{N_U} \mathbf{F}_c\left(u_i, u_j, l\right) \qquad (2)$$

Next, the conditional probability of a certain label given a transition is obtained as

$$\mathbf{P}_C\left(c \middle| u_i \rightarrow u_j, l\right) = \mathbf{P}_T\left(u_j \middle| u_i, l, c\right) / \sum_{c=1}^{N_C} \mathbf{P}_T\left(u_j \middle| u_i, l, c\right) = \mathbf{Q}_c \qquad (3)$$

and the result of the second normalization leads to an activation matrix $\mathbf{Q}_c$ of the same dimensionality as $\mathbf{F}_c$

Recognition. When recognizing utterances, keyword activations are obtained as sums of those elements of $\mathbf{Q}_c$ that are activated by the lagged acoustic units in the test sequences:

$$A(c,t) = \frac{1}{L} \sum_{l \in \boldsymbol{l}} \mathbf{Q}_c\left(u_t, u_{t+l}, l\right) \qquad (4)$$

where L is the total number of lags that was used on the time instant t (at the beginning and end of the sequence not all lags can be used).

In this work (see also [11]), the keyword model activations are smoothed by using a moving average filter of the length of 25 windows followed by a recursive decay with transfer function $H = 1/\left(1 - \left(1 - 1/\gamma\right) z^{-1}\right)$ where $\gamma = 6$. The resulting smoothed activation curves $A_S(c,t)$ show how much each word model is activated at each time instant in the recognized signal. The recognized word on each time instant is the model with the highest activation score (the winning model). Word segment boundaries are obtained from the locations where the winning word model changes. Figure 1. shows an example of a recognized utterance and the discovered word boundaries.

In order to measure model performance, we calculate how many of the N ground truth keywords are correctly detected for each test utterance. Finally we divide the number of correctly detected words by the number of all keywords over the whole test set. In this paper we select the recognized keywords as the models with the highest activation peaks. If we know that in the utterance there are N keywords to be recognized, the winning models from local maxima of the activation curves are searched, and the N highest unique models are selected. The minimum allowed distance between two peaks is set to 10 windows. Note that in previous experiments considering the basic CM algorithm, word hypotheses are created by summing model activations over all time windows and selecting the models with the highest accumulated activations [11]. For DCM, it is crucial to keep track of the locations of winning word models, and the recognition based on local maxima preserves them.

In the conventional CM technique (e.g. [11]), word models are learned by associating all acoustic transitions in an utterance to all present labels. In the above example, during the training utterance "Smiling **daddy has** the **fish**", acoustic transitions corresponding to label "fish" are thus equally updated to the correct label **fish**, as well as incorrect labels **daddy** and **to have**. However, due to varying sets of labels across training utterances, word models converge towards the correct acoustic transitions (= cross-situational learning).

2.2. Dynamic CM

If the correct locations of keywords in the training utterances were known (as in supervised learning), intuitively, the amount of noise in learned acoustic models could be reduced by training concept matrices only around the approximate locations of the keywords. Since this information is not initially given to the learner, we investigate the hypothesis that when noisy word models are gradually learned, the learner can approximate keyword locations in the training utterances and weight learning on the corresponding acoustic features.

In these experiments, on every training trial, we first recognize the training utterance (equations 2-4). From the smoothed activation curves, we get a list of winning models - for each time window $winner(t) = \text{argmax}_c(A_S(c,t))$. If any of the labels ***c*,** known to exist in the utterance, are found in the set of winners, the acoustic models of these labels are updated more strongly (here by experimentally promising double activation) surrounding their winning time windows. Mathematically, in DCM the term *a* in the update equation (1) becomes:

$$a = \begin{cases} 2, & if\ c \in \{winner(t-s), ..., winner(t+s)\} \\ 1, & otherwise \end{cases} \qquad (5)$$

where *s* is a spreading parameter defining how far the strengthening effect of a winning label spreads in time. If $s = 0$, the double update occurs only at the locations where the model to be updated wins. With larger values of *s* the update rule starts to also weight the transitions towards and away from the winning models. Figure 1 Illustrates the updating procedure of DCM as well as the spreading parameter *s*.

3. Experiments

3.1. Data preparation

The keyword recognition performance of CM and DCM are compared on Y2 and TI datasets. Y2 consists of a total of 50 keywords of which one to four occurred in each sentence. We randomly selected 2000 sentences of each of the four speakers in the training set and the remaining 397 sentences in the test set, resulting in 8000 training utterances and 1588 test utterances. TI consist of 8623 training utterances, each with one to seven spoken digits from a set of 11 digits in total ("oh", "zero", "one", "two", …). The original test set of 8700 utterances is reduced to 5455 by deleting utterances that have any digit occurring more than once in order to simplify the keyword recognition. Without this simplification, we should allow the detection algorithm to accept several local maxima of a single model's activation, possibly leading to more than one detection of a keyword on its only occurrence.

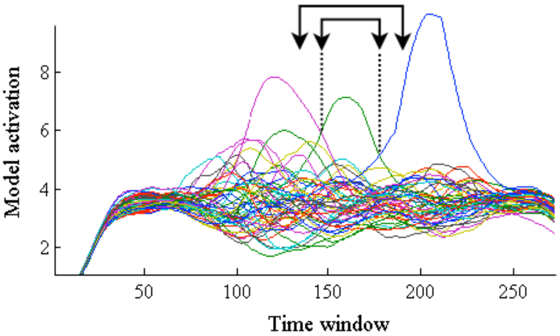

Figure 1. *Model activations for the recognized sentence "She* ***sees*** *a* ***blue eagle****" after training with 2000 training utterances. The range inside the dashed lines shows the time range where the word model "blue" wins. The larger range shows the winning area spread with parameter s, where the model "blue" is trained with double activation.*

All utterances are downsampled to 16 kHz, and transformed to MFCC-features with a window length of 32 ms and step size of 10 ms. Following the procedure in [11], in Y2 we use 12 MFCC coefficients plus log energy of each window as a feature vector. The weights of the energy and the first coefficient are attenuated by factor 0.3. In TI, only the 12 MFCC coefficients are used, with the same attenuation on the first coefficient. Additionally, on TI only, cepstral mean and variance normalization was performed on the MFCC vectors.

Both datasets were individually vector quantized into sequences of integers. VQ codebooks were obtained by randomly taking 15,000 MFCC vectors from the training dataset and clustering them into 128 (Y2) or 150 (TI) clusters using k-means clustering. In both experiments we used lags ***l*** = {-15, -14, …, 14, 15} and an experimentally found, non-optimized spreading parameter of 8. In each experiment the algorithm was run 10 times so that vector quantization and randomization of training data ordering were reapplied with every run. Additionally, in Y2, training and test sets were randomized for each run following the description above.

3.2. Experimental conditions

The performance of the two models is tested with the noiseless original datasets as well as with added ambiguity ("noise") either in the keyword labels given per utterance (*noisy labels*) or in the utterances themselves (*noisy utterances*).

Noisy labels. For each utterance the learner is given extra labels, of which none corresponds to any keyword in the utterance. This is done in order to test the algorithm's robustness to learn word models in situations where additional labels (meanings) not related to heard speech are present. In Y2 dataset we add three (low noise, LaL) and 15 (high noise, LaH) additional labels to the set of given labels ***c***. In TI, in the low noise condition we add three additional labels and in the high noise condition we add so many labels that the total number of labels per training utterance becomes 10 (i.e. only the lack of one label from the total set of 11 brings necessary ambiguity to make models converge).

Noisy utterances. Extra acoustic information is added to each training utterance, keeping the original set of labels unchanged. For both Y2 and TI, extra speech is selected for each training utterance by randomly selecting a number of utterances from the training data, not including any words corresponding to the labels given for the current training utterance, and concatenating the VQ sequence of the training utterance with VQ sequences of the additional data until a desired sequence length is reached. In TI data, in the low noise condition (UtL), we add a noise sequence of length 170 VQ indices (≈1.7 seconds), roughly duplicating the length of the training sequence (mean training sequence length = 172.6, SD = 78). In the high noise condition (UtH) we use a noise sequence length of 510 indices. In Y2 data, following the average length, in the low noise condition we use a noise sequence of length 275, and in the high noise condition a length of 825 windows.

3.3. Results

The simulation results are presented in Table 1, and in Figures 2 (noiseless conditions) and 3 (noisy conditions). DCM is seen to significantly outperform CM in both noiseless datasets after the full training data. Importantly, DCM is seen to converge towards the correct word models faster than the original CM algorithm. With low amounts of noise either in the given set of meanings or utterances, DCM generally keeps on outperforming CM. Interestingly, on Y2, three additional labels for each training utterance improves the keyword recognition accuracy significantly for both CM (Wilcoxon rank sum test, $W = 55$, $p < 0.01$) and DCM ($W = 55$, $p < 0.01$) when compared to the normal training labels. A possible explanation is that giving a few extra labels helps to train background noise and carrier words more evenly into all models and later average out their effect in test utterances.

With high amount of noise in either the utterances or the sets of keywords, CM starts to outperform DCM. Inspection of activation curves of recognized utterances provides a possible explanation. In CM, carrier sentence information is better averaged on all word models, suggesting that when the amount of noise in the utterances increases it becomes more likely that DCM starts to update some of the given word models more strongly towards often occurring noisy, non-meaningful parts of speech. Using a threshold on activation strength or duration when selecting the winners of equation (5) might help to solve these problems, and is left for future research.

Experimenting with the technique shows that the best performance is achieved when extra weighting applies to a model only surrounding the model's winning location and the extra weight is a fairly large coefficient that is not proportional to the model's activation score or its probability at those moments. For comparison, if all given models per utterance are updated with an addition corresponding to activation scores at every time instance ($a = 1 + A_S(c,t)$ in (5), cf. [17]) the model does not reach even normal CM accuracy (one run, Figure 3, dashed line). It thus seems beneficial for the learner to make a hard decision if the heard acoustic segment corresponds to any of the given labels, and update the model accordingly. Also, during training it is important that the winning models to be weighted win the competition between all possible labels **C** and not only between the given utterance-related labels ***c***.

Table 1. *Keyword recognition accuracies (%) and their standard deviations over 10 runs. * shows where the model's performance is significantly better (Wilcoxon rank sum test, p < 0.05) than the other model's.*

Data	Model	Norm	LnL	LnH	UnL	UnH
TI	DCM	**94.77*** **(0.23)**	**94.83*** **(0.24)**	77.45 (4.14)	**94.61*** **(0.22)**	**49.06** **(3.76)**
TI	CM	93.63 (0.30)	93.33 (0.19)	**88.75*** **(0.30)**	92.83 (0.24)	47.55 (0.75)
Y2	DCM	**92.39** (0.33)	93.85 (0.26)	90.50 (0.40)	**91.82*** **(0.31)**	86.42 (0.74)
Y2	CM	91.97 (0.29)	93.85 (0.19)	**91.61*** **(0.44)**	91.40 (0.33)	**86.50** **(0.37)**

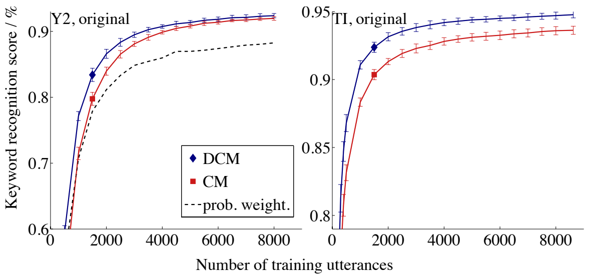

Figure 2. *Keyword recognition accuracy for the original Y2 database (left) and TI database (right).*

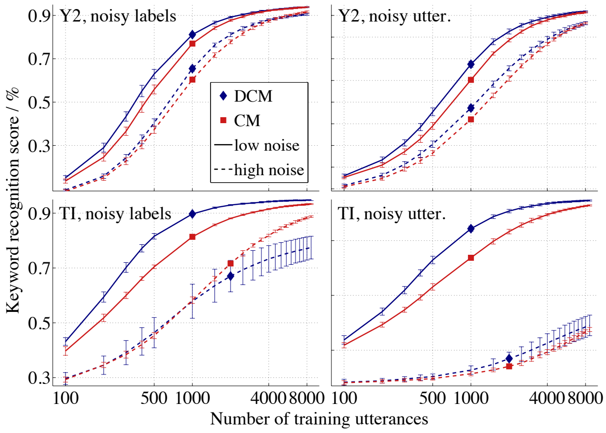

Figure 3. *Keyword recognition accuracy for the noisy datasets.*

4. Conclusions

This paper introduces an active training method for incremental weakly supervised learning, improving an existing passive learning algorithm (CM [11]). The new model, DCM, is evaluated in a keyword recognition task using real acoustic utterances. In active training, every training utterance is recognized with the word models learned thus far, and model update is affected by the recognition result, weighting the update on the regions where the given keywords are detected. Our results indicate that cognitively plausible, active online processing of training data, may lead to better word recognition performance as well as faster convergence than when training data passively or in batch mode.

Our simulations suggest that active update proportional to the model's activations (as with segmented speech in [17]) does not improve the passive CM model when using real speech signals (see Figure 3). Instead, detection of winning words and non-linear update is needed to improve performance. This suggests that cognitively plausible learners may benefit from making hard decisions about possible detection of objects/events, and in the case of positive detection focus on the details of the event in order to refine the recognizers.

5. Acknowledgements

This research was funded by the ETA Graduate School of Aalto University, Finland, the ERC starting grant project ABACUS, grant number 283435 and the Academy of Finland. The authors would like to thank Bart de Boer, Sabine van der Ham and Bill Thompson for insightful comments.

6. References

[1] J. R. Saffran, R. N. Aslin and E. L. Newport, “Statistical learning by 8-month-old infants,” Science, vol. 274, no. 5294, pp. 1926-1928, 1996.

[2] F. R. McInnes and S. J. Goldwater, “Unsupervised extraction of recurring words from infant-directed speech,” Proceedings of the 33rd Annual Conference of the Cognitive Science Society, Boston, Massachusetts, pp. 2006–2011, 2011.

[3] S. Pinker, Learnability and cognition. Cambridge, MA: The MIT Press, 1989.

[4] C. Yu and L. B. Smith, “Rapid word learning under uncertainty via cross-situational statistics,” Psychological Science, vol. 18, no. 5, pp. 414-420, 2007

[5] H. Van hamme, “HAC-models: a novel approach to continuous speech recognition,” in INTERSPEECH 2008 - 9th Annual Conference of the International Speech Communication Association, September 22–26, Brisbane, Australia, Proceedings, pp. 2554–2557, 2008.

[6] J. Driesen, L. ten Bosch and H. Van hamme, “Adaptive non-negative matrix factorization in a computational model of language acquisition,” 10th Annual Conference of the International Speech Communication Association, September 6-10, Brighton, UK, Proceedings, pp. 1731–1734, 2009.

[7] M. Versteegh, L. Ten Bosch and L. Boves, “Active word learning under uncertain input conditions,” in t*he 11th Annual Conference of the International Speech Communication Association,* pp. 2930-2933, 2010.

[8] J. Driesen and H. Van hamme, “Modelling vocabulary acquisition, adaptation and generalization in infants using adaptive Bayesian PLSA,” Neurocomputing, vol. 74, no. 11, pp. 1874-1882, 2011.

[9] T. Altosaar, L. ten Bosch, G. Aimetti, C. Koniaris,, K. Demuynck and H. van den Heuvel, “A speech corpus for modeling language acquisition: CAREGIVER,” Proceedings of the International Conference on Language Resources and Evaluation, May 19-21, Valletta, Malta, 2010.

[10] M. Sun, Constrained Non-negative Matrix Factorization for Vocabulary Acquisition from Continuous Speech, PhD Thesis, Katholieke Universiteit Leuven, Belgium, 2012.

[11] O. Räsänen and U.K. Laine, A method for noise-robust context-aware pattern discovery and recognition from categorical sequences. Pattern Recognition, 45, 606–616, 2012.

[12] D. Yurovsky, C. Yu & L. B. Smith, “Competitive processes in cross-situational word learning,” Cognitive Science, 37, pp. 891–921, 2013.

[13] E. M. Markman, J. L. Wasow, & M. B. Hansen, “Use of the mutual exclusivity assumption by young word learners,” Cognitive psychology, vol. 47, no. 3, pp. 241–275, 2003.

[14] C. C. Cheng, F. Sha, and L.K. Saul, “A fast online algorithm for large margin training of continuous density hidden Markov models”, 10th Annual Conference of the International Speech Communication Association, September 6-10, Brighton, UK, Proceedings, pp. 668-671, 2009.

[15] L. Phillips, and L. Pearl, “Less is more in Bayesian word segmentation: When cognitively plausible learners outperform the ideal,” 34th Annual Conference of the Cognitive Science Society, August 1-4, Sapporo, Japan, Proceedings, pp. 863-868, 2012.

[16] P. Liang, and D. Klein, “Online EM for unsupervised models,” In: Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the Association for Computational Linguistics, pp. 611-619, 2009.

[17] A. Fazly, A. Alishahi and S. Stevenson, S. “A Probabilistic Computational Model of Cross‐Situational Word Learning,” Cognitive Science, vol. 34, no. 6, pp. 1017-1063, 2010.