Unsupervised Hierarchical Temporal Abstraction by Simultaneously Learning Expectations and Representations

Katherine Metcalf and **David Leake**
Computer Science Department, Indiana University
{metcalka, leake}@indiana.edu

Abstract

This paper presents ENHAnCE, an algorithm that simultaneously learns a predictive model of the input stream and generates representations of the concepts being observed. Following cognitively-inspired models of event segmentation, ENHAnCE uses expectation violations to identify boundaries between temporally extended patterns. It applies its expectation-driven process at multiple levels of temporal granularity to produce a hierarchy of predictive models that enable it to identify concepts at multiple levels of temporal abstraction. Evaluations show that the temporal abstraction hierarchies generated by ENHAnCE closely match hand-coded hierarchies for the test data streams. Given language data streams, ENHAnCE learns a hierarchy of predictive models that capture basic units of both spoken and written language: morphemes, lexemes, phonemes, syllables, and words.

1 Introduction

The ability of artificial agents to interpret and understand their surroundings depends on their world models. Their capacity to learn models from sensory data depends on their ability to make sense of continuous data streams by learning discretized concept representations and deriving concept hierarchies. In this paper, *concepts* are defined as predictively useful temporal abstractions of patterns in *data streams*, where data streams are data containing sequences of statistically regular patterns presented in a streaming manner.

Event Segmentation Theory (EST) [Zacks *et al.*, 2007; Kurby and Zacks, 2008; Radvansky and Zacks, 2014] hypothesizes a process for human concept formation from sensory inputs. EST holds that people manage their world models using expectation failures. If a person is using the correct model, incoming observations should match the expectations derived from the model; failures of expectations reveal flaws in the model or that a different model should be applied. This paper proposes a new algorithm for unsupervised concept learning in streaming data domains, inspired by EST, that uses the notion of transient expectation failures as an indicator of the need for model revision. In particular, it forms representations of patterns in an input stream (its concepts), at multiple levels of abstraction, generates expectations from those concepts, and learns when expectation failures reveal that the current concepts are no longer relevant. This process could apply, for example, learning to recognize a hierarchy of units in language understanding, such as morphemes, lexemes, phonemes, syllables, and words, to support an expectation-driven understanding process.

The proposed algorithm, *ENHAnCE* ("Expectation driveN HierarhicAl Concept lEarning"), uses a shallow gated-recurrent neural network (GRNN) as its base prediction mechanism for learning and maintaining a statistical model of inputs from a data stream, following existing computational models of event segmentation [Reynolds *et al.*, 2007; Metcalf and Leake, 2017]. *ENHAnCE* extends such models with the capacity for segmenting input streams at multiple levels of temporal abstraction and learning a decomposable hierarchy of discrete, temporally extended concepts. Thus *ENHAnCE* performs unsupervised learning of discrete concept hierarchies and builds the foundation for a hierarchical world model. *ENHAnCE* uses expectation failures simultaneously to guide concept boundary recognition and to generate of a hierarchy of concept representations.

ENHAnCE was evaluated on language data because language streams are an instance of streaming data with temporal dependencies for which previously-defined concept hierarchies can be used as a reference for assessing performance. The existing language concept hierarchy (*e.g.*, characters, phones, syllables, and words) was used as ground truth to assess learning performance. The results show that, using prediction error alone as the signal for concept formation, *ENHAnCE* can learn concept hierarchies closely matching existing concept hierarchies with high accuracy and without any explicit supervision.

The following sections review the relevant background and define the necessary vocabulary; discuss related work; detail the specifics of *ENHAnCE*; present the experimental design used for evaluation; and discuss results. The paper closes by sketching potential areas for future work.

2 Background

There is a long tradition of AI research on the role of expectations and use of expectation failures—discrepancies detected between expectations and observations—to guide processing,

including areas such as early work on natural language understanding (*e.g.*, [Martin, 1989; Schank and Leake, 2002]) and recent work on goal reasoning (*e.g.*, [Aha, 2018]). Much of that work processes input that has already been segmented to assign structure, using pre-existing semantic knowledge. However, processing streaming sensory data may require segmentation for which pre-existing semantic knowledge is insufficient. This motivates the development of data-driven processes for learning the needed structures.

Deep neural network research has made great strides for learning features and representations, enabling prediction quality surpassing humans in some domains (*e.g.,* [He *et al.*, 2015; Silver *et al.*, 2017]). However, unsupervised discretization (*i.e.,* event segmentation) and concept learning remain open challenges for network methods. To address this, we turn to Psychology and Cognitive Science for inspiration, specifically to EST [Radvansky and Zacks, 2014].

According to EST, people learn to process streaming input by first learning a low-level predictive model, and then learning hierarchies of concepts, where *concept* refers to a summarized representation of the underlying temporal pattern [Newtson, 1973]. The learned predictive model is used to maintain beliefs about which concepts are being observed, supporting, *e.g.,* an expectation-driven understanding process. When a person's expectations are violated in a consistent way, observations leading up to and following the violation are mentally marked as belonging to two different concepts.

3 Related Work

Several approaches have been proposed for learning structure from streams of sensory information. Gumbsch *et al.* (2017) present a method for learning event taxonomies from continuous sensorimotor information. The model is learned to minimize the amount of free energy. It can learn conceptual structure at multiple levels of precision, and is used to understand how actions impact and change the world. Mohseni-Kabir *et al.* (2018) present a method for learning a hierarchy of actions and action primitives through demonstration and narration of the demonstration; it depends on the narration to detect action boundaries. Ha *et al.* (2015) present a method for learning hierarchical knowledge that is grounded both in vision and language. The method is grounded in incremental graph construction where concepts higher in the graph are more abstract than those at lower levels. Lee *et al.* (2016) take the approach of modulating between fast-changing local behaviors and slow-changing global patterns to learn a deep model for abstract concepts that correspond to locations, sub-locations, and activities. The model is updated and maintained via online learning and relies on a dual memory architecture to alleviate the effect of "catastrophic forgetting" sometimes observed in deep networks [Goodfellow *et al.*, 2013].

The design of ENHAnCE has key similarities with multi-scale recurrent neural networks (MRNN) [Schmidhuber, 1992; El Hihi and Bengio, 1996; Koutnik *et al.*, 2014], although it learns models for a different purpose. MRNNs are premised on some parts of the world changing rapidly and being sensitive to precise local timing relative to others that change more slowly [El Hihi and Bengio, 1996]. Taking advantage of this, multi-scale RNNs chunk hidden units into groups that update according to different schedules, which are managed via hyper-parameters [El Hihi and Bengio, 1996; Koutnik *et al.*, 2014; Bahdanau *et al.*, 2016], learned dynamic variables [Schmidhuber, 1992; Chung *et al.*, 2015; Chung *et al.*, 2016], or via learning the latent hierarchical structure [Chung *et al.*, 2016]. The hierarchical MRNN is able to learn hierarchical structure from the sequences without any explicit information about the locations of temporal boundaries.

The approach presented in this paper is most closely related to hierarchical voting experts (HVE) [Miller and Stoytchev, 2008a; Miller and Stoytchev, 2008b]. HVE extends the voting experts (VE) algorithm [Cohen *et al.*, 2007], which first learns the frequency of sequences of size n, and then segments streams of tokens into chunks that minimize internal entropy and maximize boundary entropy. HVE operates over sequences of any token type and constructs complex tokens out of those constructed by lower-level VE algorithms. Complex tokens may be composed of a sequence of characters or other complex tokens. Each level in the VE hierarchy segments the tokens presented to it, chunks those tokens that have been segmented together, and passes the sequence of chunks onto the next level to be further segmented. All tokens and chunks of tokens correspond to some meaningful, discrete unit in the original sequence, *e.g.* a word. Unlike HVE, *ENHAnCE* is able to operate on continuous observations and does not require *a priori* knowledge of the number of levels.

Previous work introduced unsupervised methods for event segmentation [Zacks *et al.*, 2007; Metcalf and Leake, 2017; Franklin *et al.*, 2019]. The Zacks *et al.* model was trained to predict the next observation and used expectation failures (SSE) to trigger memory updates. Metcalf and Leake (2017) built upon the Zacks *et al.* (2007) GRNN architecture by replacing the externally set threshold with a reinforcement learned (RL) policy. The policy observed the GRNN's SSE, controlled the gate's behavior, and was rewarded based on the SSE. Their RL+GRNN architecture demonstrated it is possible to learn the gating mechanism for identifying event boundaries. Furthermore, the RL+GRNN architecture demonstrated that it is possible to learn the gating policy without encoding any knowledge about the true location of event boundaries into the reward function.

4 *ENHAnCE*

ENHAnCE builds a predictive model of its observations and exploits that model to learn to segment its observations into a hierarchy of temporal concepts. The learning objective is to derive concepts such that, when a known concept is being observed, the model can predict (1) the **next observation(s)** and (2) the **next concept**. As an example of a task for which such prediction is useful, consider building a model of financial articles for expectation-driven natural language understanding. In this domain, references to interest rates are common. Consequently, observation of the word "interest" might predict that the next word will be "rates" and that the current concept

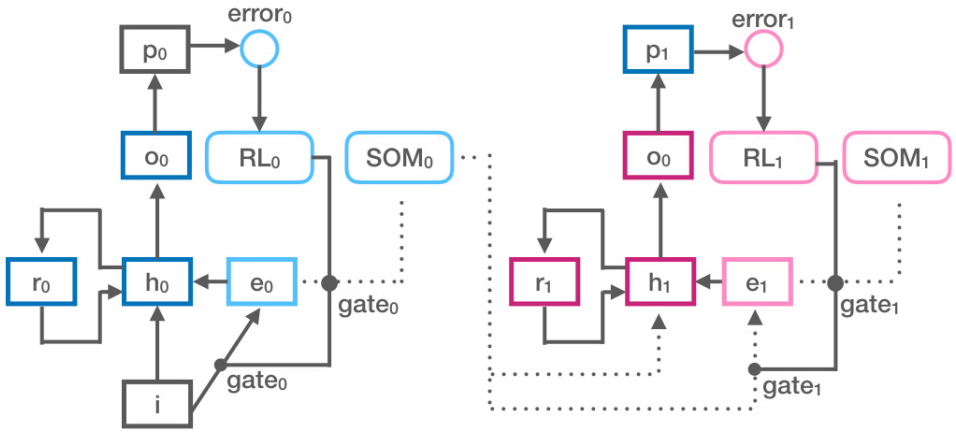

Figure 1: Iterative learning architecture for *ENHAnCE*, consisting of SRNN, RL+GRNN, and SOM structures: (1) the dark blue SRNN layer is learned, (2) the light blue RL+GRNN structure is added and learned, (3) light blue SOM is learned to cluster event states into concepts, (4) the dark pink SRNN is added and learned, (5) the light pick RL+GRNN structure is added and learned, and (6) the light pink SOM to learn event representations is added and learned. Additional SRNN, RL+GRNN, and SOM structures can be added until desired depth is reached.

is interest rates. Such a multi-level expectation-driven understanding process has been applied to NLP using pre-defined concept hierarchies [Martin, 1989]. The task of *ENHAnCE* is to generate suitable hierarchies automatically.

Figure 1 depicts how *ENHAnCE* learns in an iterative, bottom-up fashion, which forces it to learn a hierarchy that has at its lowest level concepts that cannot be decomposed into smaller, meaningful units. This aligns with theories about how humans develop complex concepts [Reynolds *et al.*, 2007]. Initially a simple recurrent neural network (SRNN) is used to learn a predictive model of the observations. The SRNN contains a single hidden state and a single context (recurrent) state, as illustrated in Figure 1 (dark blue). The model learns to predict its next observation based on current observations and its "memory" and is updated based on the error in its predictions. The SRNN is learned until convergence (operationalized as two successive gradients within 10^{-12}). Learning until convergence prior to adding additional architectural components encourages the SRNN to have some degree of stability before its output are used to learn more abstract temporal patterns.

Once the SRNN stabilizes, it is modified to include a world state embedding and a gating mechanism, resulting in a RL+GRNN layer (Figure 1, light blue). The role of the embedding and gating mechanism is to reduce prediction error by maintaining a stable representation of observations. The RL gating mechanism learns to encode information about the SRNN's error; it learns to take actions based on how "confused" the SRNN is about its next predictions. Both the world state and the gating mechanism are learned using the same prediction error signal used to learn the initial SRNN.

The RL+GRNN is learned until convergence (same criterion as RL+GRNN) at which point concepts are derived from the RL+GRNN's world state embeddings using a Growing Self-Organizing Map (GSOM) [Dittenbach *et al.*, 2000] to cluster the representations (Figure 1, light blue). A GSOM was selected for this task because it can dynamically select to add a node to the network to account for new concepts. The GSOM nodes summarize the world states and are treated as prototypical representations of world state embeddings. The nodes continue to be updated over the course of the lifetime of the GSOM. The addition of the GSOM completes the first level of concept learning. After the first level of concepts has been learned, another level is introduced. A second RL+GRNN layer is learned using the same strategy as for the first. An initial SRNN learns a temporally extended predictive model of the world (Figure 1, dark pink); instead of taking as input direct elements of the observation stream, it takes as input the world state embedding maintained by the previous layer along with the associated GSOM prototypes. It is tasked with predicting the previous layer's next event state. The second RL+GRNN layer receives input from the previous layer whenever the previous layer's gating mechanism indicates that a new concept is being observed. The layer is evaluated based on how well it can predict the prototype of the concept the previous layer will observe next. As a result, layers higher in the RL+GRNN hierarchy make predictions about which concept will be observed next. The higher into the hierarchy the RL+GRNN is, the more temporally extended its predictions are, and the further into the future the layer's predictions are targeted.

As with the first layer, once the SRNN is learned and has stabilized, RL+GRNN architectural components are added (Figure 1, light pink). Once the added world state component and the gating mechanism stabilize, the world state embeddings are clustered using the GSOM to learn prototypical world states for the layer (Figure 1, light pink).

Additional RL+GRNN layers are learned and stacked upon the first two layers following the procedure outlined above until either of two termination conditions. First, if a desired depth is known *a priori*, learning is terminated by halting the addition of levels once the desired depth is reached. Otherwise, RL+GRNN layers continue to be added and evaluated until added layers cease to converge. At such a point, the top of the concept hierarchy can be considered to have been reached as there are no longer strong statistical regularities to

Data Source	Coding One	Coding Two
Saffran Infant	char. → Morse word → phons.	phon. → int. code
1984	char. → Morse word → phons.	phon. → int. code

Table 1: Combinations of data sources and coding techniques used to evaluate *ENHAnCE*.

be learned. Once the last RL+GRNN layer has been added to the hierarchy, the set of world state prototypes learned through clustering with the GSOMs is the concept hierarchy. Tracking when the gating mechanisms trigger at each layer in the RL+GRNN hierarchy provides hierarchical annotations for the observations identifying when each concept at each level in the hierarchy is observed.

5 Training for Evaluation

In the configuration for evaluation, the model was trained iteratively one level at a time following a three-step process: (1) learn the SRNN, (2) learn the RL+GRNN, and (3) learn the GSOM. The levels in the model were learned iteratively so that each subsequent level could learn from stable input signals. In addition, learning one level at a time reduces the complexity of the gradient estimate helping to make back-propagation more stable.

The SRNN was learned to make one step predictions with a memory of what it has seen, but no explicit representation of the concept it is seeing; whereas the GRNN used a memory of what it had seen and an explicit summary representation of the type of concept it was seeing when making predictions. The GRNN's event representation was controlled by the gating policy learned to minimize the GRNN's loss.

At the first level, prediction error was measured according to the L_1 norm between the predicted next observation and the true next observation. At all later levels, the L_2 norm was used to compute the error between the predicted next concept and the correct GSOM node for the concept. The error functions were selected based on their ability to minimize the SRNN's error measured as SSE.

The architecture of the SRNN and the GRNN followed that of previous work [Zacks *et al.*, 2007; Metcalf and Leake, 2017] except for one change. Instead of a sigmoidal activation on the input layer a linear activation was used. This was chosen because it better minimized the SRNN's error. Back-propagation was handled using the Adam optimizer. The model was implemented using Tensorflow (1.2.0).

Also following the previous work, the RL policy based gating mechanism was learned using Expected Sarsa [Sutton and Barto, 2015; Metcalf and Leake, 2017]. The reward function was designed to encourage the policy to trigger the gate such that the GRNN's error was minimized. The reward at each time step was equivalent to the negative value of the GRNN's loss, *i.e.,* $-SSE$. The input to the policy was the ratio of the GRNN's current loss and its average loss:

$$state_t = \frac{SSE_{t-1}}{SSE_{avg}}, \tag{1}$$

where SSE_{t-1} is the last observed error in the GRNN and SSE_{avg} is the average error in the GRNN computed using the moving average [Zacks *et al.*, 2007]. Both the state and the reward match those used in Metcalf and Leake (2017).

At each step in the learning process, each component was learned until convergence. For the SRNN, the GRNN, and the gating mechanism's policy, the convergence condition was:

$$||\mathbf{Loss}_t - \mathbf{Loss}_{t-1}||_2 \leq 1e^{-12}, \tag{2}$$

where $\mathbf{Loss}_t$ and $\mathbf{Loss}_{t-1}$ are the current and previous loss, respectively. The GSOM was considered to converge when the ratio of variance within any node relative to the variance in the entire data set was less than 0.05.

6 Evaluation

The primary evaluation goal was to assess the ability of *ENHAnCE* to segment continuous observations into meaningful discrete concepts at multiple levels of temporal abstraction. *ENHAnCE* was evaluated on text- and speech-based data streams, using existing data streams for which concept hierarchies were already available for use as ground truth for each level of temporal abstraction (*i.e.,* characters, phonemes, syllables, words, etc.). The specific data sources were selected to allow for a direct comparison to experimental results on HVE [Miller and Stoytchev, 2008a; Miller and Stoytchev, 2008b]. The first data source was the first 30,000 words (4748 unique words) in George Orwell's *1984*. The second was the sequence of 90 words (8 unique words) used in Saffran's infant speech comprehension experiment [Saffran *et al.*, 1996]. The *1984* data contained roughly 12,000 unique syllables and 26 unique characters. The Saffran infant data set contained 12 unique syllables and 13 unique characters. The text data was pre-processed by removing all non-alphabetic characters and converting all characters to lowercase.

6.1 Encoding the Data

To assess the robustness of the methods on different input representations, each data source was coded according to two different coding mechanisms [Miller and Stoytchev, 2008a; Miller and Stoytchev, 2008b]. The coding techniques were: (1) *character to Morse code* and (2) *word to phonemes + phoneme to numeric code*. In the *character to Morse code* coding, each alphabet letter was replaced by its Morse code counterpart. Dots and dashes were represented by integer ones and zeros, respectively. Each individual 1 and 0 was treated as a single observation. The phoneme coding was designed to mimic variable length phone articulation, represent that there are multiple sample observations of a phone, and to evaluate *ENHAnCE* in a continuous domain. The *word to phonemes* coding mapped words to sequences of phonemes according to the CMU Pronouncing Dictionary. The CMU Dictionary represents over 125,000 words with 39 different text-based phoneme representations. Any word not in the CMU dictionary was excluded; such words were rare and mostly proper nouns. The *phoneme to numeric code* technique mapped the CMU Dictionary's character-based phoneme sequences to continuous observation sequences. Each phoneme was mapped to a sequence of

GRNN Prediction Error						
Data	**Encoding Type**					
	Morse code			**Phonemes**		
	L1	L2	L3	L1	L2	L3
Infant	.021	0.26	0.48	0.300	0.311	1.572
1984	0.24	0.33	0.55	0.34	0.43	1.11
Number of Learned Concepts						
Data	**Encoding Type**					
	Morse code			**Phonemes**		
	L1	L2	L3	L1	L2	L3
Infant	16	10	8	14	11	9
1984	28	11823	3255	42	10732	4862
Similarity to True Next Concept						
Data	**Encoding Type**					
	Morse code			**Phonemes**		
	L1	L2	L3	L1	L2	L3
Infant	-	0.76	0.89	-	0.401	1.032
1984	-	0.58	0.67	-	0.26	0.92

Table 2: *GRNN Error & Learned Concepts* - Reports the average prediction error in the GRNN, the number of nodes/prototypical concepts learned by the GSOM, and the similarity of predicted next concepts to the true next concept per data source and encoding type. Performance is reported per-level in the RL+GRNN hierarchy. Similarity to the next concept is not reported at Level 1, as no concept representation feeds into the first level.

observations whose values were randomly sampled according to a normal distribution. Each phoneme was assigned a unique mean, μ, on the range $[-0.5, 0.5]$ such that phonemes with similar articulations had similar means. There were 4 features per observation and the number of observations was determined by $4 \cdot |p|$, where $|p|$ is the number of characters used by the CMU Dictionary to encode the phoneme. For example, the phoneme 'o' in 'odd' is encoded as 'AA' therefore its numeric representation contained 8 observations each of length 4. The combination of data sources and coding techniques is outlined in Table 1.

The coded data were presented to the GRNN in a streaming fashion. For example, when observing the word 'odd,' coded phonetically as 'AA D,' the network received as input the numeric vector for 'AA,' predicted what it would observe next (*e.g.,* 'D'), the RL gating mechanism made a decision about whether to open or close the gate, and observed the error in its prediction.

6.2 Evaluation Criteria

For each data set and encoding, *ENHAnCE* was evaluated for three capabilities: (1) minimizing GRNN prediction error, (2) segmenting observations near ground truth concept boundaries, and (3) producing meaningful concept representations.

Minimizing GRNN prediction error: Error in the network was measured as the sum squared error (SSE) between the predicted next observation and the true next observation. Robust weights and effective control of the gating mechanism should improve the predictive power of the GRNN and reduce its overall error. Consequently, the level of GRNN prediction error evaluates both the quality of the learned weights and of learning by the RL gating mechanism.

Segmentation near ground truth boundaries: The ability to trigger the gate near ground truth boundaries indicates whether the RL gating mechanism is able to segment continuous observations corresponding to the known underlying concepts. For example, if the gate is triggered at the point where the numeric code for 'AA' transitions to the numeric code for 'D' then the RL+GRNN model has learned to recognize when the concept underlying the network's input has changed. The quality of gate triggering was evaluated according to the average number of observations between a gate trigger and the closest true boundary—scaled by the true length of the concept—(average distance), the rate at which gate triggers lined up with true boundaries (accuracy), the rate at which true boundaries were triggered on (hit rate), and an F-score summarizing the hit rate and accuracy.

Producing meaningful concept representations: A concept representation is considered to be meaningful to the extent that it can be used to predict what will be observed next. A concept representation is predicatively useful if the next level in the hierarchy of RL+GRNNs can use it to predict the next concept. The correctness of a predicted next concept is measured as its distance (euclidean) from the GSOM node closest to the true next event representation. The model is evaluated relative to each level in the hierarchy.

7 Results and Discussion

Table 2 reports the total average loss, both within concepts and at concept boundaries. We observed that at each level, prediction error between boundaries was lower than it was at boundaries. On average, the difference between the GRNN errors was 1.023. The total average error in the GRNNs increased and the difference in error signals between boundary points versus at boundary points became less pronounced at higher levels in the hierarchy. The performance of the model decreases, but not substantially so, as the complexity of the input observations and the concepts increases.

Comparing the performance of the gating mechanism across levels in the concept hierarchy shows that each level involves its own difficulties and results in different gating mechanism behaviors. The performance of the gating mechanism across metrics and the location of gate triggers, especially for the phoneme coding, indicates that *ENHAnCE* is able to learn concepts corresponding to phonemes or characters, syllables, and words. For example, at the first level in the model, the gating mechanism is able to hit many of the true event boundaries (HR). However, the model tends to over estimate the number of true event boundaries (Acc.). At the second level, the gating mechanism almost always triggers on true event boundaries, but misses some. However, it is likely that the missed boundaries are near misses based on the average number of observations between the triggered and true event boundaries. The ability to learn at higher levels in *ENHAnCE* indicates that mistakes a lower levels did not completely block the learning of concepts at higher levels.

Whereas HVE struggled to handle observations of sequences structured as Morse code, which contains common, shared subsets, the performance of *ENHAnCE* is less im-

Morse code													
	Model	**L1**				**L2**				**L3**			
		Dist.	Acc.	HR	F	Dist.	Acc.	HR	F	Dist.	Acc.	HR	F
Infant	*ENHAnCE*	0.197	0.53	0.743	0.63	0.21	0.48	0.63	0.54	0.94	0.41	0.56	0.47
1984	*ENHAnCE*	0.89	0.52	0.78	0.62	1.02	0.51	0.48	0.49	1.42	0.34	0.39	.36
	HVE	-	0.444	0.358	0.397	-	-	-	-	-	0.107	0.095	0.100
Phonemes													
	Model	**L1**				**L2**				**L3**			
		Dist.	Acc.	HR	F	Dist.	Acc.	HR	F	Dist.	Acc.	HR	F
Infant	*ENHAnCE*	0.2	.82	.98	.88	0.075	0.95	0.73	0.81	0.23	0.724	0.85	0.789
1984	*ENHAnCE*	0.089	0.88	0.95	0.91	0.064	0.98	0.89	0.93	0.12	0.93	0.91	0.92
	HVE	-	-	-	-	-	-	-	-	-	0.808	0.806	0.807

Table 3: *Segmentation* - The average distance, accuracy, hit rate, and F-score between gate triggers and true concept boundary points. Average distance reports average number of observations between a gate trigger and a true boundary, accuracy reports how many gate triggers overlap with true boundaries, hit rate reports the number of true boundaries triggered at, and F-score summarizes accuracy and hit rate. Performance is reported per-level in the RL+GRNN hierarchy and, where possible, compared against the reported HVE results. HVE does not report average distance and only reports word segmentation for phonemes and character and word segmentation for Morse code.

paired. *ENHAnCE* maintains a representation of what it has seen (in the recurrent layer and the event representation) and makes predictions based on how surprising the current observation is relative to previous observations. The prediction error is in a sense a measure of familiarity; the network is better able to predict observations that are part of patterns the network has seen frequently. In contrast to HVE, *ENHAnCE*'s performance is not perturbed by the fact that true sub-sequences do not have lower internal entropy than false sub-sequences.

Overall, as with HVE [Miller and Stoytchev, 2008a; Miller and Stoytchev, 2008b], *ENHAnCE* performed better on more natural representations of language. This is true for both the encoding type (*i.e.* Morse code vs. phoneme) and the data source (*i.e. 1984* vs. Infant Experiments). Performance was considerably better on the phoneme encoding than on the Morse code encoding. This is reasonable as Morse code requires the knowledge of true boundary points to decipher (phonemes do not) and, therefore, does not include the information theoretic context humans require to segment streams of data. On the phoneme-based encoding, *ENHAnCE* also performed better on *1984* than on the Infant experiment data source whereas *ENHAnCE* performed better on the Infant data for the Morse code encoding. This difference may initially seem counter-intuitive, as *1984* is more complex. However, *ENHAnCE* operates on points of expectation failures. As the Infant experiment data source was simpler and composed of a more regular pattern, we expect fewer expectation failures for *ENHAnCE* to operate over.

8 Conclusion and Future Work

This paper has presented a domain-independent method for detecting meaningful patterns in streams of data and learning a hierarchy of concepts from those patterns. Results demonstrate that for the test domains and encodings, (1) *ENHAnCE* is able to learn, without explicit supervision, behaviors that minimize errors in predictions about future observations, (2) that it is able to identify concepts and concept boundaries near ground truth concepts across levels in the hierarchy, and (3) that the learned concepts are meaningful in that they are predictively useful to higher levels in the model.

Future work includes evaluation for additional data sets and baselines, further investigation of the relationship between model levels and ground truth levels, and study of the performance contributions of particular steps. Longer-term work includes incorporating a feedback loop to bring information about higher hierarchy levels into the decision process at lower levels (to provide greater context for low-level decisions), and incorporating the model into one-shot learning—using *ENHAnCE*'s prediction mechanisms to identify previous familiar concepts, and, based on them, construct predictions relevant to the new concept. This moves beyond evaluating the immediate predictive usefulness of the learned concepts to evaluate the power of the representations for more extended reasoning tasks.

Acknowledgments

This work is supported by the Indiana University Precision Health Initiative. The authors thank Barry Theobald, Russ Webb, and Nick Apostoloff for their comments and contributions to model design and evaluation.

References

[Aha, 2018] David Aha. Goal reasoning: Foundations, emerging applications, and prospects. *AI Magazine*, 39(2):3–24, 2018.

[Bahdanau *et al.*, 2016] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE, 2016.

[Chung *et al.*, 2015] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075, 2015.

[Chung *et al.*, 2016] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[Cohen *et al.*, 2007] Paul Cohen, Niall Adams, and Brent Heeringa. Voting experts: An unsupervised algorithm for segmenting sequences. *Intelligent Data Analysis*, 11(6):607–625, 2007.

[Dittenbach *et al.*, 2000] Michael Dittenbach, Dieter Merkl, and Andreas Rauber. The growing hierarchical self-organizing map. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 6, pages 15–19. IEEE, 2000.

[El Hihi and Bengio, 1996] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.

[Franklin *et al.*, 2019] Nicholas Franklin, Kenneth A Norman, Charan Ranganath, Jeffrey M Zacks, and Samuel J Gershman. Structured event memory: a neuro-symbolic model of event cognition. *BioRxiv*, page 541607, 2019.

[Goodfellow *et al.*, 2013] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[Gumbsch *et al.*, 2017] Christian Gumbsch, Sebastian Otte, and Martin V Butz. A computational model for the dynamical learning of event taxonomies. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*, pages 452–457, 2017.

[Ha *et al.*, 2015] JungWoo Ha, Kyung-Min Kim, and Byoung-Tak Zhang. Automated construction of visual-linguistic knowledge via concept learning from cartoon videos. In *AAAI*, pages 522–528, 2015.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[Koutnik *et al.*, 2014] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.

[Kurby and Zacks, 2008] Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008.

[Lee *et al.*, 2016] Sang-Woo Lee, Chung-Yeon Lee, Dong-Hyun Kwak, Jiwon Kim, Jeonghee Kim, and Byoung-Tak Zhang. Dual-memory deep learning architectures for lifelong learning of everyday human behaviors. In *IJCAI*, pages 1669–1675, 2016.

[Martin, 1989] Christopher Martin. Case-based parsing. In C. Riesbeck and R.C. Schank, editors, *Inside Case-Based Reasoning*, chapter 10, pages 319–352. Lawrence Erlbaum, 1989.

[Metcalf and Leake, 2017] Katherine Metcalf and David Leake. Modelling unsupervised event segmentation: Learning event boundaries from prediction errors. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*, 2017.

[Miller and Stoytchev, 2008a] Matthew Miller and Alexander Stoytchev. Hierarchical voting experts: An unsupervised algorithm for hierarchical sequence segmentation. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pages 186–191. IEEE, 2008.

[Miller and Stoytchev, 2008b] Matthew Miller and Alexander Stoytchev. Hierarchical voting experts: An unsupervised algorithm for segmenting hierarchically structured sequences. In *AAAI*, pages 1820–1821, 2008.

[Mohseni-Kabir *et al.*, 2018] Anahita Mohseni-Kabir, Changshuo Li, Victoria Wu, Daniel Miller, Benjamin Hylak, Sonia Chernova, Dmitry Berenson, Candace Sidner, and Charles Rich. Simultaneous learning of hierarchy and primitives for complex robot tasks. *Autonomous Robots*, pages 1–16, 2018.

[Newtson, 1973] Darren Newtson. Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, 28(1):28, 1973.

[Radvansky and Zacks, 2014] Gabriel A Radvansky and Jeffrey M Zacks. *Event cognition*. Oxford University Press, 2014.

[Reynolds *et al.*, 2007] Jeremy R Reynolds, Jeffrey M Zacks, and Todd S Braver. A computational model of event segmentation from perceptual prediction. *Cognitive Science*, 31(4):613–643, 2007.

[Saffran *et al.*, 1996] Jenny R Saffran, Richard N Aslin, and Elissa L Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

[Schank and Leake, 2002] Roger C. Schank and David Leake. Natural language understanding: Models of Roger Schank and his students. In *Encyclopedia of Cognitive Science*, pages 189–195. Nature Publishing Group, London, 2002.

[Schmidhuber, 1992] Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[Sutton and Barto, 2015] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 2015.

[Zacks *et al.*, 2007] Jeffrey M Zacks, Nicole K Speer, Khena M Swallow, Todd S Braver, and Jeremy R Reynolds. Event perception: a mind-brain perspective. *Psychological bulletin*, 133(2):273, 2007.