2017

A Model of Event Knowledge

Jeffrey L. Elman

Ken McRae
kenm@uwo.ca

A Model of Event Knowledge

Jeffrey L. Elman (jelman@ucsd.edu)
Department of Cognitive Science, 9500 Gilman Drive
La Jolla, CA 92093 USA

Ken McRae (kenm@uwo.ca)
Department of Psychology, Social Science Centre
London, ON N6A 5C2 Canada

Abstract

We present a connectionist model of event knowledge that is trained on examples of sequences of activities that are not explicitly labeled as events. The model learns co-occurrence patterns among the components of activities as they occur in the moment (entities, actions, and contexts), and also learns to predict sequential patterns of activities. In so doing, the model displays behaviors that in humans have been characterized as exemplifying inferencing of unmentioned event components, the prediction of upcoming components (which may or may not ever happen or be mentioned), reconstructive memory, and the ability to flexibly accommodate novel variations from previously encountered experiences. All of these behaviors emerge from what the model learns.

Keywords: events; schema; scripts; prediction; recurrent connectionist model

Introduction

We know many things about the world. How that knowledge is organized, its content, and how it is stored, accessed, and learned have been the subject of semantic memory research for some time. A long and rich tradition of scholarship has produced a relatively stable set of theoretical constructs that are used for discussing this kind of knowledge, including categories, concepts, and features.

But people also possess another type of knowledge that has been long recognized as extremely important, although it is less clearly understood. This is knowledge about common situations and events, and has been referred to by a range of names, including pragmatic knowledge and world knowledge. Such knowledge appears to serve multiple purposes. It guides our own behavior, and helps us interpret the behavior of others. We use this knowledge to anticipate the consequences of events as they unfold. We use this knowledge extensively in language understanding to make inferences about unstated components of situations that may be incompletely described.

Bartlett (1932) was one of the first psychologists to talk about the role of such knowledge in memory. Later, in the 1970s and 1980s, cognitive psychologists such as Bransford and colleagues demonstrated that event knowledge is important in encoding and retrieving details about situations. Garrod and Sanford, among many others, showed that this kind of knowledge supports inferences in language comprehension. One assumption that appears to be shared (though was often implicit) was that the use of world/pragmatic/event knowledge in language comprehension occurred at late stages in processing. In large part this reflected theoretical assumptions of the time in linguistics and psycholinguistics, but it is also true that the typical experimental tasks used at the time were off-line, and did not lend themselves to tracking real-time incremental processing.

Over the years, there have been a number of attempts to formalize this kind of knowledge, giving rise to mechanistic explanations involving *frames* (Minsky, 1974), *scripts* (Schank & Abelson, 1977), *schema* (Norman & Rumelhart, 1981), and *stories* (Mandler, 1984), among others. Although the core intuitions motivating these proposals were widely accepted, the actual implementations revealed a number of challenges. Templates were inherently rigid and inflexible. Yet most situations admit a large range of variation and novelty. Moreover, many situations involved blends of multiple events. Symbolic architectures did not lend themselves to dealing with such challenges. Thorny questions were raised and not satisfactorily answered: What is an event (and what is not)? What is the content and detail of event knowledge? Does event knowledge have a structure common across all event types? How is event knowledge accessed and used? How is event knowledge learned? These questions remain open to this day.

Several recent developments have encouraged cognitive scientists to focus more intensely on event knowledge and how best to model it. Our own interest arises from work in language processing using real-time measures to examine processing as comprehenders deal with incrementally presented input. There is now considerable evidence that event knowledge plays a significant role in comprehension very early in processing, indeed, guiding expectations even in advance of input being received. The time course of how this knowledge is accessed and deployed is now not only of great theoretical interest (insofar as it may constrain our theories about the cognitive architecture underlying language understanding), but has become something that can be measured empirically.

A second development has been the emergence of non-symbolic computational frameworks that demonstrate the ability to capture behaviors that simultaneously reflect awareness of global abstractions as well as sensitivity to ways in which those abstractions may be graded and affected by subregularities and even idiosyncracies. Both Bayesian and connectionist models have these qualities. Our research uses a connectionist model because they exhibit key additional capabilities. They learn by example, and they allow us to probe in (simulated) real time the dynamics of the network's responses to incrementally presented input.

In the remainder of this paper, we present a model and report a set of simulations we have conducted. We begin by explaining the design criteria that guided model development. These criteria were chosen because we believe they are needed to model processes that reflect the use of event knowledge in human behavior. We conclude by discussing what we have learned from the model, and ways in which it might guide future experimental research.

The Model

Design Criteria

The model's architecture was developed with the goal that it should have the following four properties.

Learn the components that comprise an activity. We make the assumption that events can be viewed as sequences of activities, where activities occur in the moment and are comprised of various participants, actions, and contexts. Rather than prespecifying a template for necessary or sufficient components, the model must learn which components occur and co-occur across contexts and sequences. These co-occurrences may be statistically variable, and the model must learn these (often high-order) statistical interdependences.

Learn the temporal structure of activity sequences. We also assume that the temporal structure of activity sequences that make up an event may be variable across instances of any given event type. The model must learn this temporal structure, including cases in which that structure is rigid and obligatory as well as cases in which there is a high degree of variability or optionality. The model should be able to use its knowledge of the temporal structure to anticipate likely future activities, given previously encountered sequences. These expectations should reflect both global contingencies as well as predictions that may reflect more idiosyncractic variants of an activity sequence. In human terms, the model should be able to make predictive inferences.

Learn to generalize from specific examples of events. Although the model will learn from multiple examples of a given event type, it must learn the (often graded) patterns that underly them. It must also learn subregularities and if possible, exceptions.

Fill in missing information. Both during learning and testing, the model may be exposed to activity descriptions in which some highly expected information is omitted. The model should be able to activate missing elements, as appropriate (pattern completion). In human terms, the model should be able to make elaborative inferences.

Architecture

The architecture of the model is shown in Figure 1.

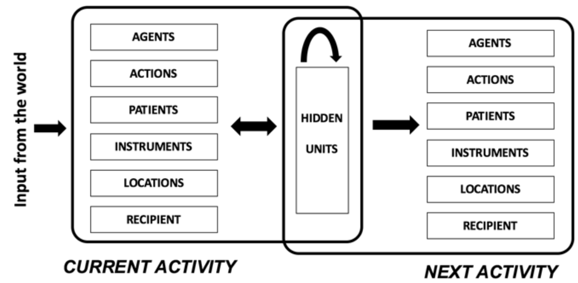

Figure 1

There is a single network, but the left and right portions play complementary roles. The left portion receives input from the world in the form of (localist) specifications of potential participants, actions, and contexts that might characterize the current activity under description. Each rectangle thus represents a number of possible inputs of the same category (*agents, patients,* etc.). It should be emphasized, however, that there is no representational status to these groups. As far as the network is concerned, every input node in all of these groups is orthogonal to every other node. If there are similarities in terms of behavior or statistics of privilege of occurrence, the network must discover them. Input nodes are fully connected to nodes in the Hidden Unit layer, and hidden units also connect (with different weights) back to input units. This use of recurrence allows the network not only to learn co-occurrence patterns among input units, but also to implement constraint satisfaction. This means that after the network has learned, it has the potential to activate missing elements in an input pattern, as appropriate. The Next Activity side of the network consists of units that are identical to the Current Activity units, but the job of the Next Activity units is to predict which activity will follow, given the sequence so far. Recurrent connections from the hidden units back to themselves are critical for this function because they provide the network with an internal representation (which must be learned) of the past that can be used for prediction. This architecture builds on elements of prior modeling that has provided a strong foundation for the present approach, including in particular Botvinick and Plaut (2004), Elman (1990), Rogers and McClelland (2004), Rumelhart, Smolensky, McClelland, and Hinton (1986), St. John and McClelland (1990), and Reynolds, Zacks, and Braver (2007).

Training and Testing

Simulations were carried out using the *rbp* package from the PDPtool simulator (McClelland, 2016). Weights in the network were initialized with random values between ±0.1 and adjusted gradually using backpropagation through time (Williams & Zipser, 2004). Training stimuli were either artificially generated activity sequences (Studies 1-3) presented one activity at a time, or sequences obtained from human norming data (Study 4). After training, testing was conducted by freezing weights and presenting the network with input sequences designed as analogs of stimuli used in human experimental paradigms. Details of the general training regime can be found in http://tatar.ucsd.edu/jeffelman/EventModelTraining.html, and details relevant to each simulation are given below.

Simulation Results

Study 1: Pattern completion and elaborative and predictive inferences

Typical language use relies heavily on interlocutors' shared knowledge. This allows speakers to omit information that is assumed to be known by the comprehender, and allows comprehenders to infer unstated information. A frequent distinction is made between elaborative inferences, which involve unstated details regarding an activity currently described, and predictive (or forward) inferences, which involve expectations about what will occur next. Bridging inferences are those in which a comprehender draws on knowledge only as needed to understand a prior statement. The extent to which, and conditions under which, such inferences are drawn remains a topic of debate. Bridging inferences are largely uncontroversial. However, whether, and under which conditions, elaborative and predictive inferences occur is still debated (for review, see Murray, Klin, & Myers, 1993). In Study 1, we first verify that the constraint satisfaction properties of the network do support inference under optimal conditions. We then examine the fragility or robustness of such inferencing because it has been claimed that discrepant data have arisen from stimulus properties and the sensitivity of behavioral measures.

Simulation 1.1 The network was trained on event sequences that ranged in length from three to six activities. The sequences might be glossed as (1) John goes to a fancy restaurant; (2) John is cutting wood in the forest, using an axe; (3) John (and other people) cut themselves accidentally with a knife, and he bleeds; (4) John (and other people) cut themselves accidentally with an axe and the wound is fatal; (5) Mary and Penny are in the library and Mary asks Penny a question, which Penny answers. Having learned these sequences, the model was then tested on novel sequences. The sequences were novel both in that they omitted critical information, and they involved new combinations of activities that the model had not encountered in the same event. Figure 2 shows activations in the Next Activity units in response to the input sequence *John is in a restaurant; John cuts himself; What happened to John?* (the query takes the form of simply presenting *John* without any specified result, so the network must fill in the information). Figure 3 shows similar activations, but in response to the sequence *John is in the forest; John cuts himself; What happened to John?*

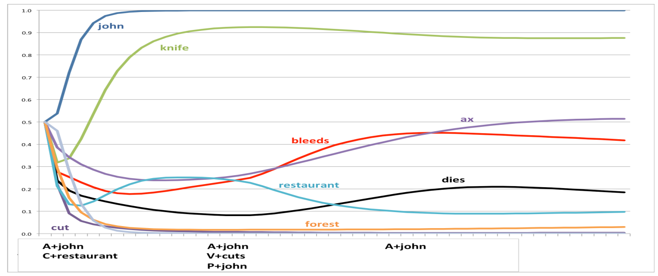

Figure 2

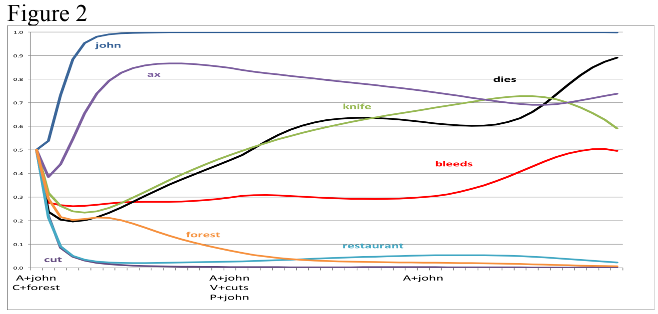

Figure 3

After receiving the input that *John* is in a *restaurant,* a knife is inferred to be present, whereas in the *forest, axe* is activated. These may be considered elaborative inferences. Then when *John cuts himself*, with no instrument mentioned, the network immediately begins to predict the result that is consistent with the instrument. These are predictive inferences. Such inferences have not always been found in humans, however. One possibility raised by Murray et al. (1993) is that failures to detect predictive inferences may result from experimental stimuli in which either the forward inference is disrupted, or it is not tested soon enough. In Error! Reference source not found., we see what happens when the discourse is disrupted by switching

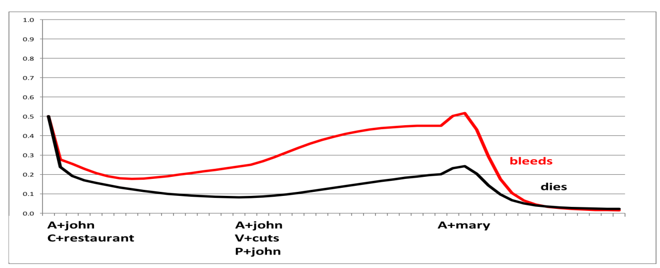

Figure 4

to a new topic

(a situation involving *Mary*) immediately after the *cutting* activity. The network begins to predict that *John* will bleed (because he is assumed to have cut himself with a knife, given the *restaurant* context). However, as soon as *Mary* is introduced, the activations of all consequences of *cut* decrease sharply. Probing for the consequence of *John* cutting himself subsequent to this topic change would show little or no evidence of the predictive inference, consistent with Murray et al.'s findings.

Simulation 1.2 One open question concerns exactly how far in the future comprehenders predict when processing incrementally presented language. In much of the experimental literature focusing on prediction in language, there has been an implicit assumption that the next word in a sequence is anticipated by comprehenders, but nothing beyond that. However, more recent findings suggest that when language is used to describe an event, comprehenders anticipate event-relevant elements even at points in the discourse where they might not be appropriate (Metusalem, Kutas, Urbach, Hare, McRae, & Elman, 2012). A simplified example of their stimuli is the short story *The crowd is in the stands. The crowd looks around. The skater goes to the podium. The audience applauds, The skater receives a ___.* Participants' brain activity was measured while reading the final noun. When an event-appropriate word, such as *medal,* was presented, the N400 amplitude was small. A final word that was completely anomalous (e.g., *bleach)* elicited a large N400. However, an word that was contextually anomalous but event-appropriate (e.g., *podium*) produced an N400 with intermediate amplitude. The authors interpreted this as evidence that event elements are activated and available even at times when they might not be immediately expected. In Figure 5, we see the network's activations in the Next Activity units throughout such a stimulus sequence.

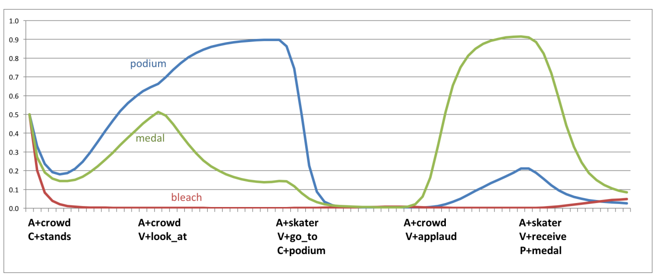

Figure 5

By the second activity, the network has already activated two event-appropriate elements, *podium* and *medal; bleach* is not activated at all. As the focus shifts in the fourth activity back to the crowd, both *medal* and *podium* are deactivated. However, near the end, the network re-activates both. The re-activation of *medal* is not surprising because it has been mentioned explicitly. However, the network has also learned that *podium* is the likely location for awarding a medal and so activates it as well, though at a lower level. There are two lessons from this simulation. First, behavioral evidence for the activation of putatively inferred event elements may depend on the timing of the probe. Second, it may be that only highly sensitive behavioral measures will reveal the presence of partially activated event elements. These elements, even if only partially activated, become more easily accessible should subsequent discourse make reference to them. This in fact was seen in Simulation 1.1.

Study 2: Novel Events and Blending

In real life, events not only exhibit variability (which the model accommodates, as we see in Figures 1 and 2) but often are combined in novel ways. Fixed templates or rigid structures are ill suited for dealing with this. In the next simulation, we test the model's ability to flexibly respond when events are combined in unusual ways.

Simulation 2.1 The model was trained on sequences that included examples of going to a restaurant (as in Simulation 1.1), and activities corresponding to a romantic relationship between two people (*John* and *Mary*), with *Mary* being married to a third person (*Bill*). Furthermore, the model was exposed to examples of aggressive behavior between various people (but not including *John* or *Bill*). In many of the latter examples, weapons are used. *Gun* is a more typical weapon, but *knives* are occasionally used. After training, the model was tested on a sequence that we might gloss as *John and Mary are at a fancy restaurant. John and Mary cut steak with a knife. Bill enters the restaurant. Bill attacks John.* Activations of relevant nodes are shown in Figure 6.

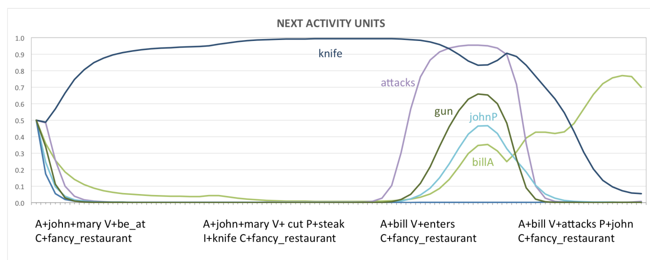

Figure 6

Two things are apparent. First, as soon as Bill enters the restaurant, the model quickly adjusts its expectations about what it predicts will happen next. Second, and more interesting, is that although the model has learned that *gun* is the most common weapon used in aggressive behavior, the presence of *knife* that was established from the outset (even prior to its mention) leads to the *knife* being the predicted instrument in this new situation. Thus, the model not only adjusts to a change in sequence structure that it has not encountered before, but it also flexibly incorporates relevant components from the first event into the second event. That is, the model produces a novel response to a situation it has never encountered before by drawing and integrating knowledge from different events.

Study 3: Priming

Studies 1 and 2 illustrate examples of priming. There is a large literature showing that event relevant information facilitates processing target elements related to that event. These include typical agents, patients, and instruments priming their event-relevant verbs, priming between event-relevant nouns, and verbs priming their event-related agents, patients, and instruments (for review, see McRae & Matsuki, 2009). The model exhibits the same behavior, not shown here because of space limitations. Instead, we demonstrate an example of priming involving second order dependencies between event elements. Bicknell et al. (2010) found that the patient that is expected to follow a given verb may depend on the agent carrying out the action. Thus, *shopper saved…* primes *money,* whereas the *lifeguard saved…* primes *person.* (Control conditions established that the priming was not directly between the agent or verb and the patient, but that it required the combination.)

Simulation 3.1 The model was trained on various examples of shoppers and lifeguards (and other people) in events in which *saving* was one of the activities. Typically, reflecting the corpus analyses carried out by Bicknell et al., *shoppers* save *money* whereas *lifeguards* save *people.* When probed with the partial description of an activity *shopper+saved* (**Error! Reference source not found.**), the model predicted *money* as the most likely patient, compared to *lifeguard+saved,* which led to greater activation of *person* (**Error! Reference source not found.**). However, we also see that that there is an asymmetry in the responses, such that at later stages in processing, *lifeguard+saved* results in an increased activation of *money* (though still lower than *person*).

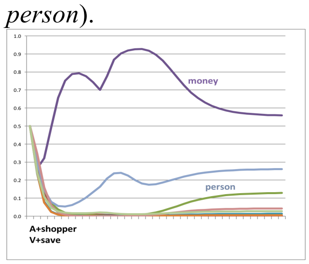

Figure 7

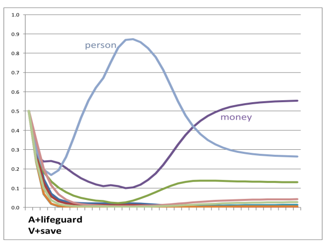

Figure 8

This reflects asymmetries in the training data that mirror asymmetries in corpus analyses, that is, that *save* is overall more commonly associated with *money* than with *people.* We might test the model's predictions (to our knowledge, as of now untested) by testing whether the timing of the patient probe leads to different degrees of facilitation, depending on when the probe was presented.

Study 4: Learning from Human Data

In the previous simulations, we used training sets that were designed by hand. The design of the training was controlled to carefully probe the network's behavior under different learning situations. This strategy is similar to that used in many human behavioral experiments. But in real life, people's knowledge of events results from experiences that may involve considerably greater variability. Consolidating such experiences and making sense of commonalities, subregularities, and exceptions is a challenge. Furthermore, temporal structure may vary considerably not only between different event types but even within a single event type. For example, there may be some parts of an event in which the ordering of activities is consistent and even obligatory (eggs must be broken before they are fried), whereas activity sequences in other parts of the event may be optional (one might make coffee before making eggs, or after). To investigate these issues, we conducted a norming study to sample people's knowledge of types of events.

Norming study 4.1 We used 81 events, drawing on prior literature that has used stimuli that describe events and situations. Some of these events have clear goals and outcomes (e.g., *fixing a flat tire*). Other events are more situation-like, in that things happen but the goal and outcome are less clear (e.g., *going to a picnic*). Using Mechanical Turk, participants were asked to list up to 12 activities for each event. Participants saw a random subset of 10-12 of the 81 events, and each event was presented to 22-24 different participants. Table 1 shows responses from three participants for *fixing a flat tire.*

Table 1: *fixing a flat tire*

Pull over	Get out of car	Pull over
Get out of car	Loosen lug nuts	Open trunk
Open trunk	Jack up car	Get tire iron
Get spare tire	Remove lug nuts	Get spare tire
Get jack	Remove flat tire	Put on hazard lights
Remove flat tire	Put on new tire	Jack up car
Put on new tire	Tighten lug nuts	Remove lug nuts
Tighten lug nuts	Remove jack	Take flat tire off
Put flat tire in trunk		Put on new tire
		Tighten lug nuts
		Lower car

The data can be visualized using graph analysis, in which nodes represent activities and directed arrows show temporal sequence (size indicates frequency), as in Figure 9.

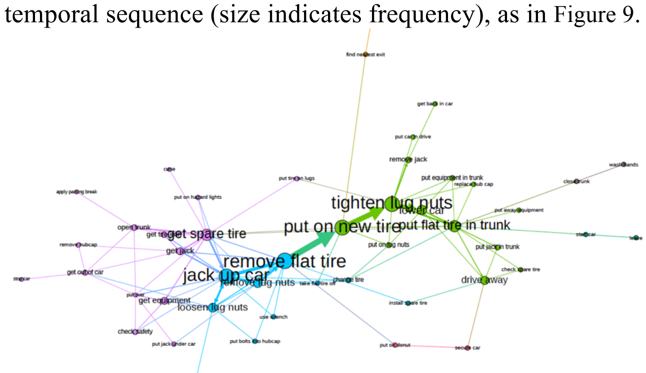

Figure 9

Some of the sequences are consistently ordered (e.g., *jack up car > remove flat tire > put on new tire*), undoubtedly reflecting causal constraints. Other sequences may be performed optionally at different times. How does the model deal with such data?

Simulation 4.1 The model was trained on the activity sequences provided by 23 participants for *fixing a flat tire.* Of particular interest is that although the model responds appropriately to the data it was trained on, its responses also incorporate what it has learned from other participants. The model thus does not slavishly reproduce the individual training data, but detects general patterns that are common across all the data.

Can the model generate activity sequences on its own? We tested this by seeding the model with a reasonable starting activity, and then using the most activated predicted elements as the subsequent input. This process iterated until the event was complete. The initial five activities in the network's self-generated sequence are shown in Figure 10 (presenting greater than five makes the figure unreadable). Notably, the network's self-generated activity sequence is not identical to any single participant's sequence. However, it is a completely reasonable abstraction of the sequencing across all participants' descriptions.

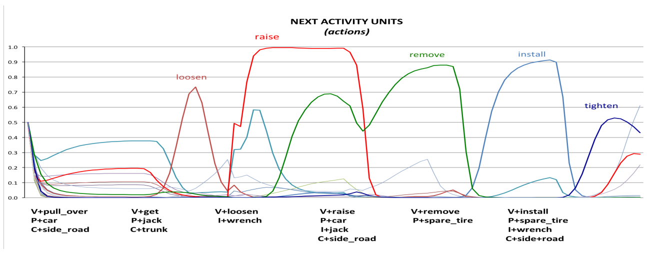

Figure 10

Discussion

Our goal was to develop a model that could learn the structure and temporal dynamics of activity sequences, as well as the co-occurrence properties of participants, activities, and contexts in those sequences. Although we might call these sequences events, the concept of event is not a primitive in the model and events are not pre-defined templates. Rather, what we might call an event is an epiphenomenal consequence of having to learn about activity sequence structure. Having done this, the architecture of the model allows it to perform pattern completion, both in the moment (supporting elaborative inferences) and across time (supporting predictive inferences). The model replicates a wide range of behavioral studies (only a few of which are described herein) for which event knowledge has been hypothesized to play a role. It also produces unanticipated behaviors that can be tested empirically to validate the model.

A great deal remains to be done. The model's inputs serve as cues to event knowledge, but the model itself does not provide those cues. Those cues must come from perceptual or motor evidence from the world as well as a language processor. Nor does the model provide an account for how these various cues can serve to alter focus on different event elements, including adjusting how the temporal contour of the event is understood (e.g., by aspect). We are guardedly optimistic that these are tractable problems and that the model we propose here provides a solid framework for understanding how people acquire, represent, and use knowledge of events in the world.

Acknowledgments

This work was supported by grants from the NIH (CHD053136) and Natural Sciences and Engineering Research Council of Canada (OGP0155704).

References

Bartlett, F. C. (1932). A Theory of Remembering. Cambridge, UK: Cambridge University Press.

Bicknell, K., Elman, J. L., Hare, M., Mcrae, K., & Kutas, M. (2010). Effects of event knowledge in processing verbal arguments. *Journal of Memory and Language, 63*(4), 489-505.

Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111*(2), 395-429.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*(2), 179-211.

Mandler, J. M. (1984). Scripts, Stories and Scenes: Aspects of Schema Theory: Hillsdale, NJ: Erlbaum.

McRae, K., & Matsuki, K. (2009). People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and Linguistics Compass*, 3, 1417-1429.

McClelland, J. L. (2017). PDPTool. Palo Alto: Stanford University. Retrieved from https://web.stanford.edu/group/pdplab/pdphandbook/

McRae, K., Hare, M., Elman, J. L., & Ferretti, T. R. (2005). A basis for generating expectancies for verbs from nouns. *Memory and Cognition, 33*(7), 1174-1184.

Metusalem, R., Kutas, M., Urbach, T. P., Hare, M., McRae, K., & Elman, J. L. (2012). Generalized event knowledge activation during online sentence comprehension. *Journal of Memory and Language, 66*(4), 545-567.

Minsky, M. (1974). *A Framework for Representing Knowledge*. Cambridge, MA: MIT Press.

Norman, D. A., & Rumelhart, D. E. (1981). The LNR approach to human information processing. *Cognition, 10*(1), 235-240.

Reynolds, J. R., Zacks, J. M., & Braver, T. S. (2007). A computational model of event segmentation from perceptual prediction. *Cognitive Science*, *31*, 613-643.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach*. Cambridge, MA: MIT Press.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in PDP models. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 2, pp. 7-57). Cambridge, MA: MIT Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

St. John, M., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence, 46*, 217-257.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation, 1*, 270-280.