Sentence Processing with Incremental Feedback

David A. Schulenburg

Technical Report – 92–51

May 27, 1992

Department of Information & Computer Science
University of California, Irvine, CA 92717 USA

An earlier version of this paper appears as *Sentence Processing with Realistic Feedback* in the *Proceedings of the International Joint Conference on Neural Networks*, Baltimore, MD, 1992.

Sentence Processing with Incremental Feedback

David Schulenburg (SCHULENB@ICS.UCI.EDU)
Department of Information & Computer Science
University of California, Irvine, CA 92717 USA

Abstract

Utilizing recurrent network topologies to produce case/role meaning representations for single sentences has become common practice in connectionist natural language processing systems. Typically, these systems train with the complete sentence meaning as the target output for the entire period that the sentence is being processed; i.e., the complete meaning is available starting with the first word of the sentence. Thus, the context feedback provided by these systems is non-incremental in that they use information about the sentence that has not yet been encountered in order to aid in the processing and learning tasks. SAIL1 is a connectionist natural language processing system which builds the sentence meaning representation incrementally, incorporating into the meaning only the information derived from words already processed.

Introduction

Utilizing recurrent network topologies to produce case/role meaning representations for single sentences has become common practice in connectionist natural language processing systems (McClelland and Kawamoto, 1986; Miikkulainen and Dyer, 1991). Typically, these systems will train with the entire sentence meaning represented as the target output while the sentence is being processed sequentially, one word at a time. Thus, the recurrent feedback (context) and/or target output (used during the backpropagation algorithm) is non-incremental; it contains information relating to the entire sentence, even before the sentence has been completely processed. Were the sentence processing task a prediction task (i.e., predict what the sentence meaning will be as early as possible), this would seem to be a viable approach.[1] Intuitively, however, one would expect a natural language system to build the meaning representation of the sentence incrementally, relying only on those words already presented to the system. Symbolic approaches to NLP work in this fashion. In this paper, I present a connectionist system utilizing a simple recurrent network which builds the sentence meaning representation in the output layer incrementally, as the sentence is processed. During training, the target output captures only that part of the entire sentence meaning represented by the words of the sentence presented up to that point.

Consider the following: The girl ate the spaghetti with the fork.[2] This sentence could be represented by the following case/role assignments:

```
[(act eat) (agent girl) (object spaghetti) (instrument fork)]
```

[1] Clearly, prediction is important in NLP as expectations from the early part of a sentence can constrain later processing. However, these constraints are typically general, e.g., the more general constraint is "physical object" rather than "John's bike."

[2] From McClelland and Kawamoto (1986).

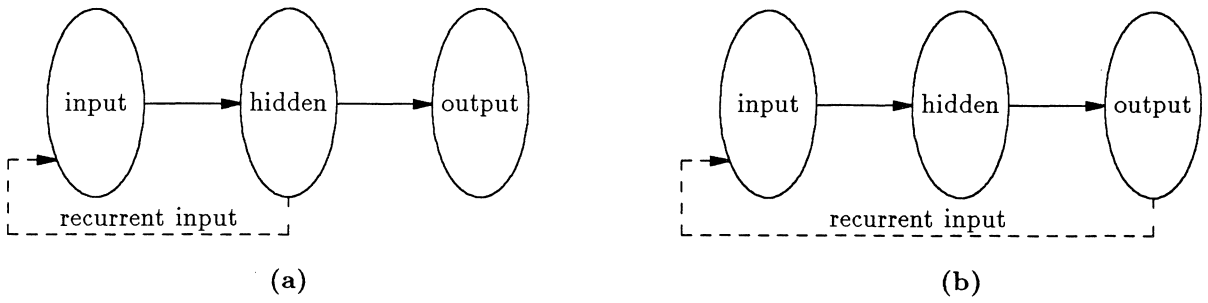

Figure 1. Two recurrent network topologies.

where there would be some indication that the act is past tense. In a typical connectionist NLP system, the network is trained to "predict" the full meaning representation of the entire sentence after seeing only the first word or phrase, e.g. The or The girl. That is, upon being presented with the, the target output, used by the backpropagation algorithm, is the entire sentence meaning representation. This implies that the full meaning representation is available to the network for training, either by a recurrent relation from the output layer or for use as the target output to compute the error during the backpropagation algorithm.

Recurrent Networks

The nature of natural language understanding (processing words sequentially to derive a meaning representation) within the connectionist framework suggests the use of a recurrent network, a network where part of the input layer consists of the output activation from the hidden layer(s) and/or the output layer. The recurrent relation can provide a context which captures the result of the processing of previous words in the sentence. Training of and processing in a recurrent network is the same as for a straight feedforward network with one exception: As each word is processed through the recurrent network sequentially, the recurrent relation (the activation of the units from the hidden or output layer) from the previous training cycle must be copied to the input layer. When the entire sentence has been processed the recurrent relation must be reset indicating that the network is ready to process the next sentence. Two types of simple recurrent network topologies are illustrated in Figure 1. The first, proposed by Elman (1990), has the recurrent relation coming from the hidden layer. This type of recurrent network does not suffer directly from the problem of building a meaning representation in the output layer non-incrementally. The hidden layer does not explicitly contain the training information found in the output layer; however, it is available implicitly via the computation of the "error" used in the backpropagation algorithm. The second type of simple recurrent network is a slightly modified version of that proposed by Pollack (1988) with the recurrent feedback coming directly from the output layer. In this topology, the non-incremental feedback problem manifests itself directly; in addition to the error backpropagation, the feedback explicitly contains the entire meaning representation. In processing sentences sequentially, then, feedback would contain information about the sentence not yet processed.

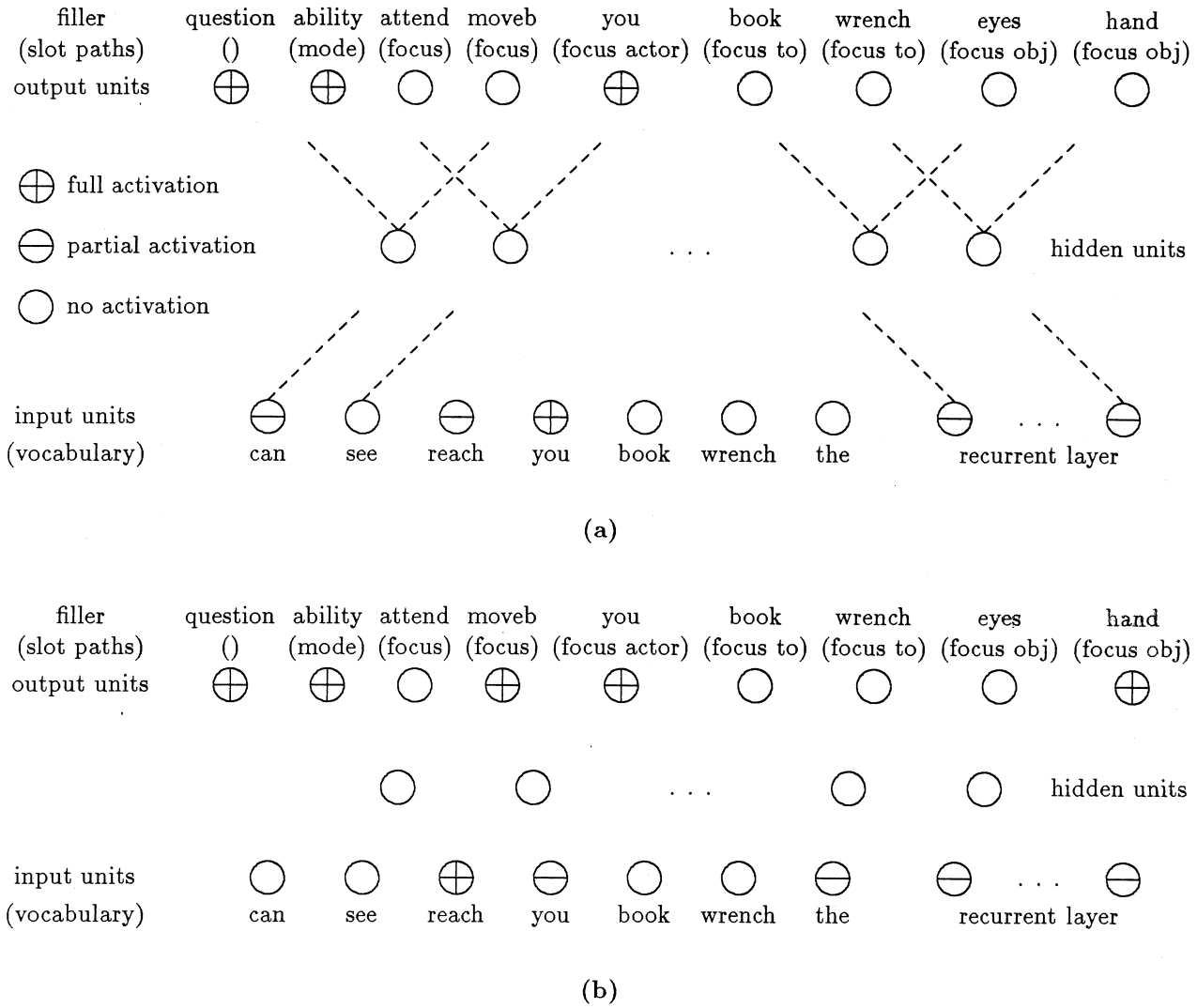

Figure 2. Example network
"Can you reach the book?"

SAIL1

SAIL1 is a connectionist natural language model which utilizes a recurrent network topology to process the words of sentences sequentially. It builds a case/role meaning representation of the sentence in the output layer incrementally, adding to the meaning representation the information from the sentence as the words are processed.

Input to SAIL1 is a sentence in English with each input unit representing an English word. SAIL1 uses a localist representation in the input layer and output layer (Figure 2). The sentence is processed one word at a time, left to right. Thus, to process the entire sentence requires n passes through the network where n is the number of words and punctuation marks in the sentence. While there is no principled reason for using a localist representation in the

Table 1. Simulation results.

feedback	recurrent layer	training epochs	total matches	non matches	output units correct	mean error[2] (test)	mean error[2] (train)
incremental	output	1500	138(92%)	13(8%)	99.741%	0.060045	0.007324
non-incremental	output	1500	111(74%)	40(26%)	99.045%	0.388422	0.190042
non-incremental	output	3000	97(64%)	54(36%)	98.608%	0.518968	0.173689
incremental	hidden	1500	130(86%)	21(14%)	98.911%	0.085628	0.006588
non-incremental	hidden	1500	102(68%)	49(32%)	98.294%	0.325612	0.195228
non-incremental	hidden	3000	132(87%)	19(13%)	99.135%	0.100205	0.008218

output layer, there is one for the input layer. Having each word represented as a unit in the input layer allows SAIL1 to provide a context when processing sentences at the word level. This context is different from that provided by the recurrent relation. The word-level context manifests itself as a "sliding window" across the words of a sentence. The sliding window is loosely based on the idea of a *Wickelfeature* (Wickelgren, 1969).[3] The word currently being processed (the focus) is at the center of the window and receives a full activation. The words on either side of the focus form the rest of the window and each receives a partial activation, currently set at one half of full activation. The focus changes as the window slides across the words of the sentence. As Miikkulainen and Dyer (1991) show, the network can be trained to learn a distributed representation over its lexicon, adding another hidden layer between the existing input and hidden layers. An approach similar to this will be added to SAIL1 in the future in which the input layer will act as a dictionary lookup to find the meaning as represented by the activation of the units in the new hidden layer.

Training of a new sentence in SAIL1's network is illustrated in Figure 2 for the sentence Can you reach the book? In SAIL1, this sentence would be represented by the following case/role assignments (in Lisp notation):[4]

```
(question mode (ability)
         focus (moveb actor (you)
                      obj (hand)
                      to (wrench)))
```

In Figure 2(a), the current focus word is you; hence, it receives a full activation in the input layer; the surrounding words, can and reach, each receive partial activation. The recurrent portion of the input layer receives the activation of the output units from the previous cycle.[5] The target output activation indicates how the sentence meaning representation is built incrementally; only the information from that part of the sentence which has been

[3] It is also similar to decay of input in other temporal connectionist systems (Jain, 1991).

[4] Each unit in the output layer represents a filler for a particular slot (defined by the "slot path") in the meaning representation, which can be mapped into a Lisp list notation. The empty path represents the head of the list; other elements of the path define the slots to which the associated filler belongs. Thus, the slot path (focus to) represents the to slot of the focus slot of the total representation.

[5] Experimentation is currently underway using the target output instead of the actual output activation as the recurrent portion of the input layer during training. Initial results indicate that the network trains more rapidly.

processed so far (i.e., Can you) is incorporated into the meaning: the first (slot path: *empty*), second (mode), and fifth (focus actor) output units. When reach becomes the focus (Figure 2(b)), we incorporate into the sentence meaning the associated information by adding full activation for the fourth (focus) and ninth (focus obj) output units. Finally, when book becomes the focus, SAIL will activate the sixth (focus to) output unit.

SAIL1 has demonstrated its usefulness by successfully generalizing to create meaning representations for novel sentences.[6] In one simulation (Table 1), the data given in McClelland and Kawamoto (1986) was used to generate training and testing data. Specifically, sentences involving the verbs ate and hit were used. As the sentences were generated, each had an independent probability of 0.9 of being included in the training data. Of the total 1330 possible sentences using this data, this simulation yielded 1179(89%) sentences in the training set and 151(11%) in the test set. The network consisted of 35 input units: one for each word of the lexicon, one for the period punctuation, and one "null" unit, used for the context supplied by the sliding window at the beginning and end of sentences. There were forty hidden units and 59 output units. A "match" is defined to be a meaning representation whose actual output unit activations (for all units) is within $\epsilon = 0.2$ of the target activations. This criteria was also used to determine the percentage of output units that were correct.

The data in Table 1 indicate that using incremental feedback gives better generalization than does a network trained using non-incremental feedback. With the recurrent relation from the output layer (rows one to three) and after 1500 training epochs, the network trained with incremental feedback was 18% more accurate than the network trained with non-incremental feedback, even though the difference in the percentage of output units that were correct (within $\epsilon = 0.2$ of target value) was very small. This latter observation would seem to indicate that the non-matches were not far off target, which is indeed the case. Many of the sentences which did not match had most of their output units reach their target values, often leaving an unfilled slot (e.g. for the verb hit, the `instrument` slot was often left unfilled). It was a very rare occurrence when a slot was incorrectly filled. Further training of the network with non-incremental feedback shows an improvement in the mean squared error for the training data but a loss of generalization on the test data indicating that the network has been overtrained; extra training here does not help. When the hidden layer supplies the recurrent relation, twice as many training epochs are required for the non-incremental feedback network to reach the level of performance of the incremental feedback network. However, this still does not match the performance given by the incremental network with feedback from the output layer.

The data in Table 1 show that the approach of SAIL1 (i.e., building the output representation incrementally) produces better results than the standard approach of trying to predict the entire output representation from the outset. One plausible explanation for this is that SAIL1's approach keeps on the right track from the beginning of sentence processing and doesn't vacillate between final meaning representations. Consider a set of sentences with some subset having different meanings but identical beginnings. With non-incremental feed-

[6] Novel sentences are those not included in a training set.

back, the network is trained on the same first word for these sentences with different target output activations. Such a network can never converge during training; it attempts to learn two different target outputs for the same input. A network trained with incremental feedback does not suffer this. The data clearly shows that building the output meaning representation incrementally in SAIL1 succeeds.

Related work

McClelland and Kawamoto (1986) present an early system which creates a case/role representation for the meaning of sentences. Because the topology of this system does not allow for sequential input, it suffers from "non-incremental input" instead of non-incremental feedback. St. John and McClelland (1990) present a system which does not create a case/role representation of a sentence meaning directly; rather, their goal is to output a "sentence gestalt" to capture the sentence meaning. The sentence gestalt is used as input along with "probes" to another network to output case/role filler pairs. When training the network, they use the entire set of probes and case/role fillers of a sentence beginning with the first sentence presented to the system. So, in this sense, the non-incremental feedback problem does exist for them. Miikkulainen (1991) discusses an extension to his CLAUSES system (Miikkulainen and Dyer, 1991). In this extended version, output is a sequence of representations capturing the meaning of the set of relative clauses of the sentence. Two networks are employed: a "phrase parser" and a "sentence parser." The phrase parser accepts the words of a sentence sequentially and is trained on individual "act fragments" (i.e., clauses). The target output, representing the meaning of the entire act fragment, clearly violates the principle of incremental feedback. Because the sentence parser does not create a meaning representation for the entire sentence but rather a sequence of representations meant to capture the meaning of the individual clauses, the incremental feedback principle does not apply.

Discussion

SAIL1 is an experiment to test the idea of incremental feedback; i.e., can a recurrent network be trained such that the output meaning representation is built incrementally? The evidence presented suggests that the answer is "yes." But why is it important to be able to build an output meaning incrementally? One major reason is that, intuitively, one would guess that this is the way that people do it. Another reason deals with the set of training and testing sentences. Current connectionist natural language processing systems rely on "grammars" for generating sentences. Given the structure inherent in these corpora, a fixed target output meaning representation seems to be a valid approach, where the meaning can "evolve" as the sentence is processed. However, these systems tend to have severe problems with sentences outside of the structure defined by their grammars. Applying such techniques to multi–sentence text would exacerbate this problem. Clearly, an approach in which the meaning is extended as new information is garnered, such as that supplied by SAIL1, is needed.

Consider a "double" recurrent network (Figure 3) where the output of one trainable network serves as the input to another. The first subnetwork processes text at the "intra–sentence" level; i.e., it builds a meaning representation for single sentences. It uses context supplied

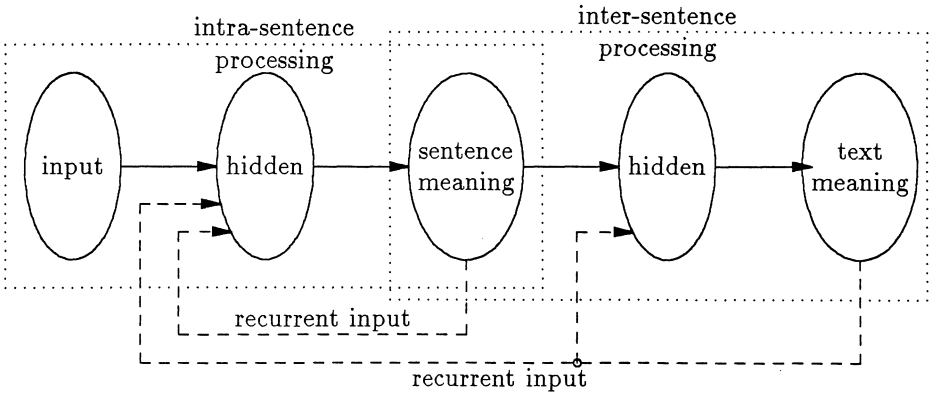

Figure 3. "Double" recurrent network topology.

by its own recurrent relation in addition to a context supplied by the second subnetwork, which assists in disambiguation at the sentence level (e.g. to help with pragmatics). This second subnetwork processes text at the "inter–sentence" level; i.e., it builds a meaning representation for multiple sentences in a body of text. In addition to being an experiment to test some ideas about connectionist natural language processing at the intra–sentence level, SAIL1 is the first step in realizing a system capable of processing multi–sentence text as exemplified by the double recurrent network. Future plans include expanding SAIL1 to the topology as shown in Figure 3 and testing that network on multi–sentence text in an effort to interpret indirect speech acts (Searle, 1975) correctly using the context supplied by the inter–sentence processing network.

Conclusion

This paper has presented a connectionist natural language processing model (SAIL1) which uses a recurrent network topology to process English sentences. The meaning representation created by the model is built incrementally. The success of SAIL1 in building the output meaning representation incrementally indicates that this approach is a viable alternative to the model in which the final output meaning representation is available during processing of the entire sentence. I have argued that, if connectionist natural language processing systems are to scale up from single sentences to multi–sentence text, incremental feedback will be required.

References

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

Jain, A. N. (1991). Parsing complex sentences with structured connectionist networks. *Working Notes, AAAI Spring Symposium Series: Connectionist Natural Language Processing.* Stanford University. 84-90.

McClelland, J. & A. Kawamoto (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. McClelland, D. Rumelhard, and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol 2: Psychological and Biological Models.* Cambridge, MA: MIT Press.

Miikkulainen, R. (1991). Parsing embedded clauses with simple recurrent networks. *Working Notes, AAAI Spring Symposium Series: Connectionist Natural Language Processing.* Stanford University. 211-215.

Miikkulainen, R., & M. G. Dyer (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science*, *15*, 343-400.

Pollack, J. B. (1988). Recursive auto-associative memory: Devising compositional distributed representations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Earlbaum Associates.

Searle, J. (1975). Indirect speech acts. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics, Vol. 3, Speech Acts.* New York: Academic Press.

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

Wickelgren, W. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, *76*, 1-15.