A Cognitive Model of Coherence-Driven Story Comprehension

Elliot Smith
School of Computer Science, University of Birmingham,
Edgbaston, Birmingham B15 2TT. United Kingdom.
email: e.smith@cs.bham.ac.uk

Abstract

Current models of story comprehension have three major deficiencies: (1) lack of experimental support for the inference processes they involve (e.g. reliance on prediction); (2) indifference to 'kinds' of coherence (e.g. local and global); and (3) inability to find interpretations at variable depths. I propose that comprehension is driven by the need to find a representation that reaches a 'coherence threshold'. Variable inference processes are a reflection of different thresholds, and the skepticism of an individual inference process determines how thresholds are reached.

1 Introduction

Recent research in psychology maintains that comprehension is 'explanation-driven' (Graesser et al., 1994) and guided by the 'need for coherence' (van den Broek et al., 1995). The comprehender's goal is construction of a more-or-less coherent representation which includes explanations for and relations between the story's eventualities. This representation is generated via inferences, which enrich the representation until it reaches the threshold specified by the comprehender's *coherence need* (van den Broek et al., 1995).

By contrast, early models of comprehension emphasised its expectation-driven nature: prediction of future eventualities, followed by substantiation of these predictions (DeJong, 1979). The inference processes described in these early models are still implemented in many contemporary systems.

One problem with these models is their failure to account for experimental evidence about inferences: predictive inferences are not generated at point x in the story, unless strongly supported by the story up to point x (Trabasso and Magliano, 1996); in addition, predictive inferences not immediately confirmed by the story after point x are not incorporated into the representation (Murray et al., 1993). While it is difficult to define 'strong support' or 'confirmation', it is clear that an overly-assumptive model does not reflect mundane comprehension.

A second problem is the failure of these models to account for differential establishment of local and global coherence. Local coherence holds between 'short sequences of clauses', while global coherence is measured in terms of 'overarching themes' (Graesser et al., 1994). McKoon and Ratcliff (1992) maintain that only local coherence is normally established during comprehension (the *minimalist* hypothesis). Others state that readers 'attempt to construct a meaning representation that is coherent at both local and global levels' (the *constructionist* hypothesis) (Graesser et al., 1994). Script-based models allow globally-coherent structures to be constructed automatically, contradicting the minimalist hypothesis; the inclusion of promiscuous predictive inferences also contradicts the constructionist hypothesis.

A third problem is that previous models deny comprehension's flexibility. This issue is sometimes side-stepped by assuming that comprehension concludes with the instantiation of one or more 'primitive' or 'top-level' patterns. Another approach is to apply lower-level patterns which account for smaller subsets of the input, but the aim is still to connect a story's first eventuality to its last (van den Broek et al., 1995).

This paper describes a model which treats inferences as *coherence generators*, where an inference's occurrence depends on its coherence contribution. Unusual inference-making, establishment of local and global coherence, and variable-precision comprehension can be described within this framework.

2 Coherence and Satisficing

A *schema* is any function which maps inputs onto mental representations. It contains slots which can be instantiated using explicit input statements, or implicit statements derived via proof or assumption. Instantiated schemas form the building blocks of the comprehender's representation. A comprehender has available both 'weak' schemas, which locally link small amounts of input (e.g. causal schemas); and 'strong' schemas, which globally link larger sections of input (e.g. scripts).

All schemas generate 'connections of intelligibility' which affect the coherence of a representation (Harman, 1986). Coherence is a common 'currency' with which to measure the benefit of applying a schema. Instead of requiring that a top-level structure be instantiated, the system instead applies schemas to produce a representation of sufficient 'value'. This process can be naturally described as *abduction*, or 'inference to the best explanation' (Ng and Mooney, 1990).

Previous natural-language abduction systems *can* form more-or-less coherent representations: for example, by halting comprehension when assumptions start to reduce coherence (ibid.). However, these systems still have a fixed 'cut-off' point: there is no way to change the criteria for a good representation, for example, by requiring high coherence, even if this means making poorly-supported assumptions. By treating coherence as the currency of comprehension, the emphasis shifts from creating a 'complete' representation, to creating a *satisficing* one. (A satisficing representation is not necessarily optimal, but one which satisfies some minimal constraint: in this case, a *coherence threshold.*)

3 Coherence-Driven Comprehension

In this section, I outline some general principles which may attenuate the performance of a comprehension system. I begin with the general definition of a schema:

$c_1, ..., c_n \rightarrow I.$

where $c_1, ..., c_n$ are the elements connected by I. The left-hand side of a schema is its *condition set*, and the right-hand side represents the *interpretation* of those conditions in terms of other concepts (e.g. a temporal relation, or a compound event sequence). During each processing cycle, condition sets are matched against the set of *observations.*

At present, I am developing a metric which measures *coherence contribution* with respect to a schema and a set of observations:

$C = (V \times U) - (P \times S)$

where C = coherence contribution; V = Coverage; U = Utility; P = Completion; and S = Skepticism. This metric is based on work in categorisation and diagnosis, and measures the similarity between the observations and a condition set (Tversky, 1977).

3.1 Coverage and Completion

Coverage captures the principle of conflict resolution in production systems. The more elements matched by a schema, the more coherence that schema imparts on the representation, and the higher the Coverage. By contrast, *Completion* represents the percentage of the schema that is matched by the input (i.e. the completeness of the match). Coverage and Completion thus measure different aspects of the applicability of a schema. A schema with high Coverage may match all of the observations; however, there may be schema conditions that are unmatched. In this case, a schema with lower Coverage but higher Completion may generate more coherence.

3.2 Utility

The more observations a schema can explain, the greater its coherence contribution. *Utility* measures this inherent usefulness: schemas with many conditions are considered to contribute more coherence than schemas with few. Utility is independent of the number of observations matched, and reflects the structure of the knowledge base (KB). In previous comprehension models, the importance of schema size is often ignored: for example, an explanation requiring a long chain of small steps may be less costly than a proof requiring a single large step. To alleviate this problem, I have made a commitment to schema 'size', in line with the notion of 'chunking' (Laird et al., 1987). Chunked schemas are more efficient as they require fewer processing cycles to arrive at explanations.

3.3 Skepticism

This parameter represents the unwillingness of the comprehender to 'jump to conclusions'. For example, a credulous comprehender (with low Skepticism) may make a thematic inference that a trip to a restaurant is being described, when the observations lend only scant support to this inference. By raising the Skepticism parameter, the system may be forced to prove that such an inference is valid, as missing evidence now decreases coherence more drastically.[1]

4 Example

Skepticism can have a significant impact on the coherence contribution of a schema. Let the set of observations consist of two statements:

enter(john,restaurant), order(john,burger)

Let the KB consist of the schema (with Utility of 1, as it is the longest schema in the KB):

$$enter(Per, Rest), order(Per, Meal), leave(Per, Rest) \rightarrow restaurantvisit(Per, Meal, Rest).$$

In this case, $C = (V \times U) - (P \times S)$, where:

$$Coverage(V) = \frac{ObservationsCovered}{NumberOfObservations} = \frac{2}{2}$$
$$Utility(U) = 1$$
$$Completion(P) = \frac{ConditionsUnmatched}{NumberOfConditions} = \frac{1}{3}$$
$$Skepticism(S) = \frac{1}{2}$$

Therefore, $C = \frac{5}{6}$, with *leave(john, restaurant)* being the assumption. If S is raised to 1, C now equals $\frac{2}{3}$, with the same assumption. Raising S makes the system more skeptical, and may prevent hasty thematic inferences.

5 Future Work

Previous models of comprehension have relied on an 'all-or-nothing' approach which denies partial representations. I believe that changing the goal of comprehension from top-level-pattern instantiation to coherence-need satisfaction may produce models capable of producing partial representations.

One issue to be addressed is how coherence is incrementally derived. The current metric, and many previous ones, derive coherence from a static set of observations. This seems implausible, as interpretations are available at any point during comprehension. A second issue is the cost of assuming various conditions. Some models use weighted conditions, which differentially impact on the quality of the representation (Hobbs et al., 1993). A problem with these schemes is the sometimes ad hoc character of weight assignment: as an antidote to this, I am currently constructing a method for deriving weights from condition distributions over the KB. This moves the onus from subjective decisions to structural criteria.

References

G.F. DeJong. 1979. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3:251–273.

A.C. Graesser, M. Singer, and T. Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395.

G. Harman. 1986. *Change in View.* MIT Press, Cambridge, MA.

J.R. Hobbs, M.E. Stickel, D.E. Appelt, and P. Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

J.E. Laird, A. Newell, and P.S. Rosenbloom. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64.

G. McKoon and R. Ratcliff. 1992. Inference during reading. *Psychological Review*, 99(3):440–466.

J.D. Murray, C.M. Klin, and J.L. Myers. 1993. Forward inferences in narrative text. *Journal of Memory and Language*, 32:464–473.

H.T. Ng and R.J. Mooney. 1990. On the role of coherence in abductive explanation. In *Proceedings of the 8th AAAI*, pages 337–342, Boston, MA, July-August.

T. Trabasso and J.P. Magliano. 1996. Conscious understanding during comprehension. *Discourse Processes*, 21:255–287.

A. Tversky. 1977. Features of similarity. *Psychological Review*, 84:327–352.

P. van den Broek, K. Risden, and E. Husebye-Hartmann. 1995. The role of readers' standards for coherence in the generation of inferences during reading. In R.F. Lorch, Jr., and E.J. O'Brien, editors, *Sources of Coherence in Reading*, pages 353–373. Lawrence Erlbaum, Hillsdale, NJ.

[1] Skepticism is a global parameter which 'weights' all schema applications. Local weights could also be attached to individual conditions (see section 5).

Tree-based Analysis of Simple Recurrent Network Learning

Ivelin Stoianov
Dept. Alfa-Informatica, Faculty of Arts, Groningen University, POBox 716, 9700 AS Groningen,
The Netherlands, Email:stoianov@let.rug.nl

1 Simple recurrent networks for natural language phonotactics analysis.

In searching for a connectionist paradigm capable of natural language processing, many researchers have explored the Simple Recurrent Network (SRN) such as Elman(1990), Cleermance(1993), Reilly(1995) and Lawrence(1996). SRNs have a context layer that keeps track of the past hidden neuron activations and enables them to deal with sequential data. The events in Natural Language span time so SRNs are needed to deal with them.

Among the various levels of language processing, a phonological level can be distinguished. The Phonology deals with phonemes or graphemes – the latter in the case when one works with orthographic word representations. The principles governing the combinations of these symbols is called phonotactics (Laver'1994). It is a good starting point for connectionist language analysis because there are not too many basic entities. The number of the symbols varies between 26 (for the Latin graphemes) and 50 *(for the phonemes).

Recently, some experiments considering phonotactics modelling with SRNs have been carried out by Stoianov(1997), Rodd(1997). The neural network in Stoianov(1997) was trained to study the phonotactics of a large Dutch word corpus. This problem was implemented as an SRN learning task – to predict the symbol following the left context given to the input layer so far. Words were applied to the network, symbol by symbol, which in turn were encoded orthogonally, that is, one node standing for one symbol (Fig.1). An extra symbol ('#') was used as a delimiter. After the training, the network responded to the input with different neuron activations at the output layer. The more active a given output neuron is, the higher the probability is that it is a successor. The authors used a so-called *optimal threshold method* for establishing the threshold which determines the possible successors. This method was based on examining the network response to a test corpus of words belonging to the trained language and a random corpus, built up from random strings. Two error functions dependent on a threshold were computed, for the test and the random corpora, respectively. The threshold at which both errors had minimal value was selected as an optimal threshold. Using this approach, an SRN, trained to the phonotactics of a Dutch monosyllabic corpus containing 4500 words, was reported to distinguish words from non-words with 7% error. Since the phonotactics of a given language is represented by the constraints allowing a given sequence to be a word or not, and the SRN managed to distinguish words from random strings with tolerable error, the authors claim that SRNs are able to learn the phonotactics of Dutch language.

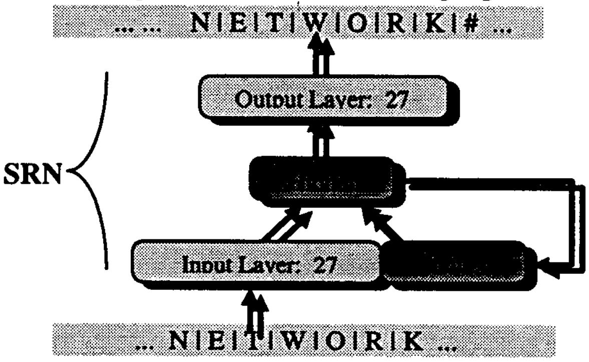

Fig.1. SRN and mechanism of sequence processing. A character is provided to the input and the next one is used for training. In turn, it has to be predicted during the test phase.

In the present report, alternative evaluation procedures are proposed. The network evaluation methods introduced are based on examining the network response to each left context, available in the training corpus. An effective way to represent and use the complete set of context strings is a tree-based data structure. Therefore, these methods are termed *tree-based analysis*. Two possible approaches are proposed for measuring the SRN response accuracy to each left context. The first uses the idea mentioned above of searching a threshold that distinguishes permitted successors from impossible ones. An error as a function of the

* for Dutch, and up to at most 100 in other languages.

threshold is computed. Its minimum value corresponds to the SRN learning error rate. The second approach computes the local proximity between the network response and a vector containing the empirical symbol probabilities that a given symbol would follow the current left context. Two measures are used: $\mathbf{L_2}$ norm and normalised vector multiplication. The mean of these local proximities measures how close the network responses are to the desired responses.

2 Tree-based corpus representation.

There are diverse methods to represent a given set of words (corpus). Lists is the simplest, but they are not optimal with regard to the memory complexity and the time complexity of the operations working with the data. A more effective method is the tree-based representation. Each node in this tree has a maximum of 26 possible children (successors), if we work with orthographic word representations. The root is empty, it does not represent a symbol. It is the beginning of a word. The leaves do not have successors and they always represent the end of a word. A word can end somewhere between the root and the leaves as well. This manner of corpus representation, termed *trie*, is one of the most compact representations and is very effective for different operations with words from the corpus.

In addition to the symbol at each node, we can keep additional information, for example the frequency of a word, if this node is the end of a word. Another useful piece of information is the frequency of each node C, that is, the frequency of each left context. It is computed recursively as a sum of the frequencies of all successors and the frequency of the word ending at this node, provided that such a word exists. These frequencies give us an instant evaluation of the empirical distribution for each successor. In order to compute the successors' empirical distribution vector $\mathbf{T^c}(.)$, we have to normalise the successors' frequencies with respect to their sum.

3 Tree-based evaluation of SRN learning.

During the training of a word, only one output neuron is forced to be active in response to the context presented so far. But usually, in the entire corpus there are several successors following a given context. Therefore, the training should result in output neurons, reproducing the successors' probability distribution. Following this reasoning, we can derive a test procedure that verifies whether the SRN output activations correspond to these local distributions. Another approach related to the practical implementation of a trained SRN is to search for a cue, giving an answer to the question whether given symbol can follow the context provided to the input layer so far. As in the *optimal threshold method* we can search for a threshold that distinguishes these neurons.

The tree-based learning examination methods are recursive procedures that process each tree node, performing an *in-order* (or *depth-first*) tree traversal. This kind of traversal algorithms start from the root and process each sub-tree completely. At each node, a comparison between the SRNs reaction to the input, and the empirical characters distribution is made. Apart from this evaluation, the SRN state, that is, the context layer, has to be kept before moving to one of the sub-trees, in order for it to be reused after traversing this sub-tree.

On the basis of above ideas, two methods for network evaluation are performed at each tree node C. The first one computes an error function $\mathbf{f^c(t)}$ dependent on a threshold $\mathbf{t}$. This function gives the error rate for each threshold $\mathbf{t}$, that is, the ratio of erroneous predictions given $\mathbf{t}$. The values of $\mathbf{f^c(t)}$ are high for close to zero and close to one thresholds, since almost all neurons would permit the correspondent symbols to be successors in the first case, and would not allow any successor in the second case. The minimum will occur somewhere in the middle, where only a few neurons would have an activation higher than this threshold. The training adjusts the weights of the network so that only neurons corresponding to actual successors are active. The SRN evaluation is based on the mean $\mathbf{F(t)}$ of these local error functions (Fig.2a).

The second evaluation method computes the proximity $\mathbf{D^c} = |\mathbf{N^c}(.)\ ,\mathbf{T^c}(.)|$ between the network response $\mathbf{N^c}(.)$ and the local empirical distributions vector $\mathbf{T^c}(.)$ at each tree node. The final evaluation of the SRN training is the mean $\mathbf{D}$ of $\mathbf{D^c}$ for all tree nodes. Two measures are used to compute $\mathbf{D^c}$. The first one is $\mathbf{L_2}$ norm (1):

$$(1)\ |\mathbf{N^C}(.)\ ,\mathbf{T^C}(.)|_{L2} = [\mathbf{M^{-1}}\,\Sigma_{x=1..M}\,(\mathbf{N^C}(x)-\mathbf{T^C}(x))^2\,]^{1/2}$$

The second is a vector multiplication, normalised with respect to the vector's length (cosine) (2):

(2) $|N^C(.), T^C(.)|_v = (|N^C(.)|\,|T^C(.)|)^{-1} \Sigma_{x=1..M} (N^C(x) T^C(x))$

where M is the vector size, that is, the number of possible successors (e.g. 27) (see Fig. 2b).

4 Results.

Well-trained SRNs were examined with both the *optimal threshold method* and the *tree-based approaches*. A network with 30 hidden neurons predicted about 11% of the characters erroneously. The same network had mean **L₂** distance 0.056 and mean vector-multiplication proximity 0.851. At the same time, the *optimal threshold method* rated the learning at 7% error. Not surprisingly, the tree-based evaluations methods gave higher error rate – they do not examine the SRN response to non-existent left contexts, which in turn are used in the *optimal threshold method.*

Discussion and conclusions.

Alternative evaluation methods for SRN learning are proposed. They examine the network response only to the training input data, which in turn is represented in a tree-based structure. In contrast, previous methods examined trained SRNs with test and random corpora. Both methods give a good idea about the learning attained. Methods used previously estimate the SRN recognition capabilities, while the methods presented here evaluate how close the network response is to the desired response – but for familiar input sequences. The desired response is considered to be the successors' empirical probability distribution. Hence, one of the methods proposed compares the local empirical probabilities to the network response. The other approach searches for a threshold that minimises the prediction error function. The proposed methods have been employed in the evaluation of phonotactics learning, but they can be used in various other tasks as well, wherever the data can be organised hierarchically. I hope, that the proposed analysis will contribute to our understanding of learning carried out in SRNs.

References.

Cleeremans, Axel (1993). *Mechanisms of Implicit Learning*.MIT Press.

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

Elman, J.L., et al. (1996). *Rethinking Innates*. A Bradford Book, The Mit Press.

Haykin, Simon. (1994). *Neural Networks*, Macmillan College Publisher.

Laver,John.(1994).*Principles of phonetics*,Cambr.Un.Pr.

Lawrence, S., et al.(1996).NL Gramatical Inference A Comparison of RNN and ML Methods. *Connectionist, statistical and symbolic approaches to learning for NLP*, Springer-Verlag,pp.33-47

Nerbonne, John, et al (1996). Phonetic Distance between Dutch Dialects. In G.Dureux, W.Daelle-mans & S.Gillis(eds) *Proc.of CLIN, pp.*185-202

Reilly, Ronan G.(1995).Sandy Ideas and Coloured Days: Some Computational Implications of Embodiment. *Art. Intellig. Review*,9: 305-322.,Kluver Ac. Publ.,NL.

Rodd, Jenifer. (1997). Recurrent Neural-Network Learning of Phonological Regula-rities in Turkish, *ACL'97 Workshop: Computational Natural language learning*, pp. 97-106 .

Stoianov, I.P., John Nerbonne and Huub Bouma (1997). Modelling the phonotactic structure of natural language words with Simple Recurrent Networks, *Proc. of 7-th CLIN'97* (in press)

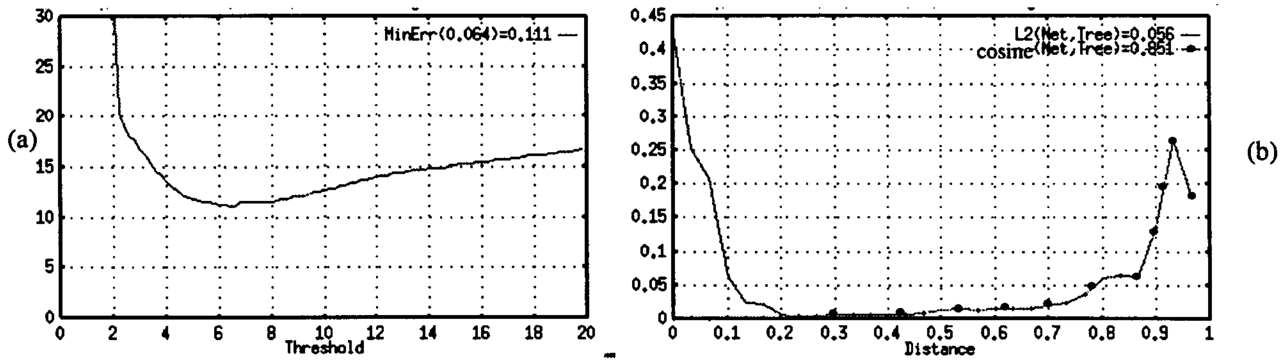

Fig.2. SRN evaluation by: (a.) minimising the error function **F(t)**. (b.) measuring the SRN matching to the empirical successor distributions. The distributions of **L₂** distance and cosine are given (see the text).