SENTENCE COMPREHENSION AS THE CONSTRUCTION OF A SITUATIONAL REPRESENTATION: A CONNECTIONIST MODEL

Stefan L. Frank

Nijmegen Institute for Cognition and Information, Radboud University Nijmegen, P.O.Box 9104, 6500 HE Nijmegen, THE NETHERLANDS, S.Frank@nici.ru.nl

ABSTRACT

When a sentence is understood, the reader or listener must have constructed a mental representation of the situation described by the sentence. The model presented here views sentence comprehension as the construction of such a 'situational representation'. The model consists of a simple recurrent neural network that transforms word sequences (i.e., sentences) into representations of the corresponding situations. These representations were developed for the DSS model of story comprehension (Frank et al., 2003). The results indicate that the network did not just perform simple word-to-situation or sentence-to-situation mappings, but also learned how new combinations of words refer to new combinations of situations.

1. INTRODUCTION

Most sentences describe a state of affairs in the world. Understanding such sentences comes down to constructing a mental representation of the state of affairs, or *situation*, referred to. According to van Dijk and Kintsch, 1983, this so-called *situational representation* is the highest of three levels of the mental representation of a text. At the lowest level, only the literal wording of the text is represented. This gives rise to a representation of the text's predicate-argument structure, which is still closely linked to the original text. The situational representation, on the other hand, is comparable to what Johnson-Laird, 1983, calls a mental model. This representation is not linguistic but based on the comprehender's experience with, and knowledge about, the world. If the asserted situation had not been perceived in the form of language, but had been experienced directly, a similar representation would have resulted (Fletcher, 1994).

Although, usually, the reader's or listener's goal is to construct a situational representation, most models of language comprehension focus on the grammatical or propositional structure of the language. World knowledge is only involved minimally, if at all. For instance, the INP model (Budiu and Anderson, 2004) views sentence comprehension as the construction of the sentence's propositional structure. The integration of this structure with world knowledge comes down to the association of a single world-knowledge fact.

In this paper, a model is presented that does not deal with propositional structures. Instead, it takes a sentence as input and transforms it into a representation of the situation described. These situational representations are taken from the Distributed Situation Space (DSS) model (Frank et al., 2003), which simulates how readers draw knowledge-based inferences when comprehending a story. In the DSS model, each situation that can occur (in a very limited world) is distributively represented by a vector in a high-dimensional 'situation space'. Section 2.2 explains how this situation space was developed. The sentence comprehension model, described in Section 2.3, takes the form of a simple recurrent neural network. A sentence is presented to it as a temporal sequence of words, and the network learns to associate such word sequences with the correct situation space vector.

Frank et al., 2005, trained an identical network to perform the same task, without very good results. Although their network successfully learned to comprehend the sentences it was trained on, generalization was rather poor. Two types of generalization are needed if the network is claimed to simulate sentence comprehension. First, the network has to be able to comprehend new descriptions of the situations it was trained on. For instance, if the model has learned to construct the situational representation of the sentence *Bob and Jilly play soccer*, it should be able to generate the same output to *Jilly and Bob play soccer*, which is a different sentence describing the same situation. Second, the model should be able to comprehend sentences describing situations on which it was not trained, but which are combinations of known situations. For instance, if the network successfully learned to comprehend the sentences *Jilly plays hide-and-seek outside* and *Jilly plays with the dog inside*, it should also be able to comprehend *Jilly plays hide-and-seek inside* and *Jilly plays with the dog outside*, even if it has not been trained on any situation in which hide-and-seek is played inside or anyone plays with the dog outside.

The Frank et al., 2005, model showed only the first type of generalization and can therefore hardly be considered a model of sentence comprehension. However, this problem may have been caused by the use of a suboptimal training regime. Here, the same network is trained on

The research presented here was supported by grant 451-04-043 of the Netherlands Organization for Scientific Research (NWO). I would like to thank Leo Noordman for commenting on an earlier version of this paper.

the same input, but using a more appropriate technique, as explained in Section 3.2. This time, the resulting network does show both types of generalization.

2. THE MODEL

2.1. Microworld and microlanguage

As argued in the Introduction, understanding language requires not only linguistic knowledge, but also world knowledge. Unfortunately, no realistically large amount of world knowledge can be implemented for a computational model to deal with. There are basically two solutions to this problem. The first is to select, for the sentence currently being processed, a small number of world knowledge facts that might be relevant to the sentence. This solution is used by the INP model (Budiu and Anderson, 2004) and the Construction-Integration model (Kintsch, 1988). The problem with this approach is that the modeler needs to choose the necessary facts for each particular sentence that is to be processed. As a result, the model is no longer fully computational. Therefore, a second solution is used here. In this approach, world knowledge is restricted to knowledge about a microworld. Only sentences describing situations in this world are processed by the model.

The microworld is taken from Frank et al., 2003. It has two characters, called Bob and Jilly, who can perform activities like 'play soccer' or 'play with the dog', and can be in different states, such as 'outside' and 'tired'. In total, any microworld situation can be constructed from boolean combinations of the 14 'basic events' listed in Table 1. Several constraints hold among these events, for instance, soccer can only be played outside, and only one game can be played at a time. Other constraints are probabilistic in nature, for instance, it is more likely for Bob and Jilly to be outside when the sun shines than when it rains.

Table 1. Fourteen basic microworld events and their intended meanings.

name	meaning
sun	The sun shines
rain	It rains
B outside	Bob is outside
J outside	Jilly is outside
soccer	Bob and Jilly play soccer
hide-and-seek	Bob and Jilly play hide-and-seek
B computer	Bob plays a computer game
J computer	Jilly plays a computer game
B dog	Bob plays with the dog
J dog	Jilly plays with the dog
B tired	Bob is tired
J tired	Jilly is tired
B wins	Bob wins
J wins	Jilly wins

Situations in the microworld can be described using the microlanguage developed by Frank et al., 2005. It has just 15 words, some of which correspond to multiple words in English: *Bob, Jilly, and, plays, is, wins, loses, soccer, hide-and-seek, a_computer_game, with_the_dog, outside, inside, tired, awake*. Using the simple grammar from Table 2, 328 different microlanguage sentences can be constructed. Note that the microlanguage has no morphology, so verbs are not inflected.

Each of the microlanguage sentences refers to a microworld situation. In the mapping from sentences to situations, some words correspond to negations of basic events: *inside* refers to the negation of 'being outside', and *awake* refers to the negation of 'being tired'. Also, *bob loses* is the situation 'J wins' and *jilly loses* refers to 'B wins'. Note that some sentences (e.g., *jilly plays soccer inside*) refer to situations that violate the microworld constraints (in this case, the constraint that soccer is never played inside).

2.2. Representing microworld situations

Knowledge of the microworld is implemented via a 'microworld description', which is a sequence of $k = 250$ example situations that follow the microworld constraints. In each situation j, each basic event p is either the case or not the case, which is indicated by the value $\nu_j(p)$. If event p occurs in situation j, $\nu_j(p) = 1$; if it does not, $\nu_j(p) = 0$.

Suppose event p would be represented by the vector $\nu(p) = (\nu_1(p), \ldots, \nu_k(p))$. That is, the j-th element of the representation of p indicates whether p was the case in the j-th example situation. Such a representation has some interesting and useful properties. First, it is easy to compute the representation for negations ($\neg p$) and conjunctions ($p \wedge q$) of basic events:

$$\nu_j(\neg p) = 1 - \nu_j(p) \tag{1}$$

$$\nu_j(p \wedge q) = \nu_j(p)\nu_j(q). \tag{2}$$

Equation 1 states that j is a situation in which $\neg p$ is the case if and only if p is not the case in situation j. Equation 2 states that $p \wedge q$ is the case in exactly those situations where both p and q are the case.[1] From these two equations follows the representation of disjunctions $(p \vee q)$:

$$\begin{aligned}\nu_j(p \vee q) &= \nu_j(\neg(\neg p \wedge \neg q)) \\ &= 1 - ((1 - \nu_j(p))(1 - \nu_j(q))) \\ &= \nu_j(p) + \nu_j(q) - \nu_j(p)\nu_j(q). \end{aligned} \tag{3}$$

Using Equations 1 to 3, the vector representation ν of any combination of basic events can be computed.

A second advantage of this representation is that the a priori and conditional probabilities that events occur in the microworld description can be computed directly from their representations. It is easy to see that the a priori probability of event p equals

$$\Pr(p) = \frac{1}{k}\sum_{j=1}^{k} \nu_j(p). \tag{4}$$

[1] It may seem more straightforward to use $\nu_j(p \wedge q) = \min\{\nu_j(p), \nu_j(q)\}$, which is equivalent to Equation 2 since each $\nu_j(p) \in \{0, 1\}$. However, for reasons that will be made clear shortly, Equation 2 is preferred.

Table 2. Grammar of the microlanguage.

S	→	NP VP
NP	→	*Bob* | *Jilly* | *Bob and Jilly* | *Jilly and Bob*
VP	→	*plays* Game [Place | *and is* State | *and* Result]
	→	*is* Place [*and plays* Game | *and* State | *and* Result]
	→	*is* State [*and plays* Game | *and* Place | *and* Result]
	→	Result [*and plays* Game | Place | *and is* State]
Game	→	*soccer* | *hide-and-seek* | *a_computer_game* | *with_the_dog*
Place	→	*outside* | *inside*
State	→	*tired* | *awake*
Result	→	*wins* | *loses*

The conditional probability that p is the case given that q is, follows from Equations 2 and 4:

$$\Pr(p|q) = \frac{\Pr(p \wedge q)}{\Pr(q)} = \frac{\sum_j \nu_j(p)\nu_j(q)}{\sum_j \nu_j(q)}. \qquad (5)$$

The vectors ν form a non-linguistic representation of situations in the microworld: Being based on a microworld description that does not make use of language, they do not reflect any linguistic or propositional structure. Instead, the vectors encode probabilistic relations among the represented situations. This means that the representation is situational in the sense of van Dijk and Kintsch, 1983.

There is one major drawback to this scheme: The length of the vectors equals k, the number of example situations in the microworld description. The current value of $k = 250$ results in reasonably short vectors, but since there is no theoretical upper bound to k, the vectors can become unpractically large. Moreover, it would be more satisfying if the size of the representations would not depend on the amount of data (i.e., the value of k) from which they are computed. What is desired, therefore, is a method of transforming the vectors ν into vectors that have a fixed (and reasonably short) length, while retaining their useful (i.e., situational) properties.

Frank et al., 2003, accomplish this by training a Self-Organizing Map (SOM; Kohonen, 1995) on the microworld description. The SOM was trained on the k input vectors $\nu_j = (\nu_j(p), \nu_j(q), \nu_j(r), \ldots)$, where $p, q, r, \ldots$ denote the 14 basic events of Table 1. That is, the j-th input vector consisted of the 14 binary values indicating which basic events are (not) the case in the j-th example situation.

During training, each of the $n = 150$ SOM cells i obtains a *membership value* $\mu_i(p) \in [0, 1]$ for each basic event p, indicating the extent to which cell i is part of the representation of event p. Figure 1 shows the SOM's membership values for two basic events. Instead of $\nu(p)$, the vector $\mu(p) = (\mu_1(p), \ldots, \mu_n(p))$ is now used as the representation of event p. This reduces the dimensionality of the representations from $k = 250$ to $n = 150$. More importantly, n is independent from k.

The vectors μ are called *situation vectors* because they represent situations in the microworld. Frank et al., 2003, show that they have the same properties as the vectors ν, expressed by Equations 1 to 5, so any microworld situation can be represented by combining vectors of basic events, and (conditional) probabilities of situations follow directly from these representations. Of course, reducing the dimensionality does throw away some of the information in the vectors ν. As a result, the probability estimates computed with vectors μ are not identical to those computed with ν. Therefore, Frank et al., 2003, do not speak of probabilities but of *belief values*, which are denoted by the symbol τ instead of $\Pr$. That is, $\tau(p|q)$ is the 'amount of belief' the system has that event p is the case given that q is the case, which is an estimate of the probability $\Pr(p|q)$. Belief values are computed similarly to probabilities, using Equations 4 and 5 where ν is replaced by μ, Pr by τ, and k by n.[2]

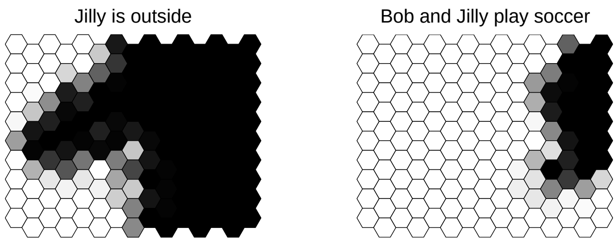

Figure 1. Representations of two basic events on the Self-Organizing Map. The gray value of a cell i for event p indicates its membership value $\mu_i(p)$. The pattern representing 'J outside' overlaps with the pattern for 'soccer', indicating that playing soccer implies that Jilly is outside.

2.3. The sentence comprehension network

Microlanguage sentences are transformed into situational vector representations by a simple recurrent neural network (Elman, 1990), the architecture of which is shown in Figure 2.

Words are represented locally at the input layer, which has one unit for each word. Since the words of a sentence are processed one-by-one, only one of the input units is active at any moment. Take, for example, the sentence *jilly*

[2] Belief values share with probabilities the useful property that $\tau(\neg p) = 1 - \tau(p)$ and $\tau(\neg p|q) = 1 - \tau(p|q)$. It is not hard to show that this would not be the case if conjunction was computed by $\mu_i(p \wedge q) = \min\{\mu_i(p), \mu_i(q)\}$.

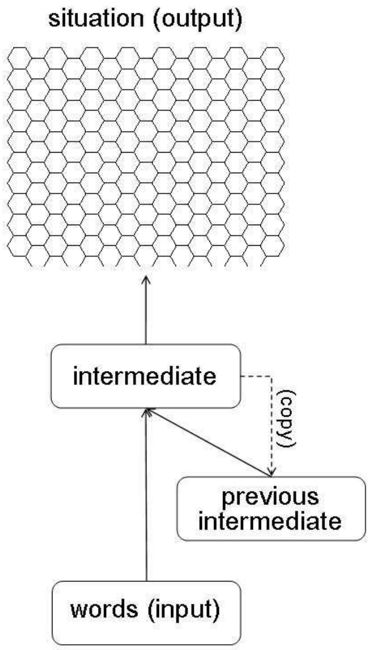

Figure 2. The sentence comprehension network. The input layer consist of 15 units, one for each microlanguage word. The intermediate level has 6 units. The situational output level has 150 units, one for each dimension of situation space. Solid arrows indicate full connectivity from one layer to the next. The dashed arrow indicates that, after processing a word, the intermediate activations are copied to the 'previous intermediate' layer. They serve as input to the intermediate layer when the next word is processed.

plays soccer. First, the *jilly* unit of the input layer becomes active. It activates the intermediate layer which sends activation to the output layer. At this moment, the output cannot be expected to be anything interesting. Next, the intermediate activations are copied to the 'previous intermediate' layer, and the *plays* input unit is activated. The intermediate layer receives this activation as well as its own previous activation. As a result, the activation pattern of the intermediate layer represents 'the sentence so far'. This pattern is propagated to the output layer which, ideally, now represents the disjunction of all situations in which Jilly plays. After copying the intermediate activations again, the *soccer* input unit is activated. The intermediate layer will come to represent the complete sentence, and the output should now be the situation vector μ(soccer).[3]

This model bears some resemblance to the ones by St.John and McClelland, 1990, and Rohde, 2002, who use neural networks with a similar recurrent architecture to process word sequences. However, those networks differ from the one presented here in the task they are trained to perform. They do not learn to transform a sentence into an independently developed representation of the described situation that is strongly integrated with world knowledge. Instead, these networks are taught to answer simple questions about the sentence. Knowledge about the world only arises to the extent that it is needed to answer these questions. For instance, the St.John and McClelland, 1990, network learns to answer *spoon* when it is asked which instrument is used in the event described by the sentence *the teacher ate the soup* (St.John and McClelland, p. 234). Considering the weak relation between sentence representation and world knowledge in those two models, it can be argued that they do not simulate sentence comprehension as it was defined in the Introduction.

[3]Note that soccer is always played by Bob and Jilly together: If she plays soccer, so does Bob. This is way there is no basic event 'Jilly plays soccer'.

3. TRAINING THE NETWORK

3.1. Training and test sentences

Of course, sentence processing only leads to the correct output representation if the network's connection weights are appropriate. These weights were set by training the network on 290 of the 328 possible sentences using the standard backpropagation algorithm of Rumelhart et al., 1986.

The other 38 sentences, listed in Table 3, were divided into two test sets. The 'old situations' test set contains 22 new sentences describing situations that were also present in the training set. For instance, the training set contained *bob is tired and outside* while the sentence *bob is outside and tired* was in the 'old situations' test set. Each sentence in this test set could be turned into a training sentence by swapping the words or phrases on either side of *and*. Since the connective *and* is commutative, this operation does not change the described situation.

The 16 sentences in the 'new situations' test set described situations that were not described by any of the training sentences. Half of the sentences in this group where about playing with the dog inside, and the other half described a situation in which hide-and-seek was played outside. Neither of these situations were presented to the network during training.

Sentences describing situations that violate the microworld constraints are treated as any other sentence. Strictly speaking, these sentences describe situations that are not impossible but only so unlikely that they never occurred in the microworld description. Since the situation vectors μ do not encode the microworld description perfectly, these highly unlikely situations do have a representation in situation space and belief values larger than 0. Note that this is not the case for the 'perfect' representations formed by the vectors ν.

3.2. Training regime

Both the network and its task are identical to the ones in Frank et al., 2005. The only difference is the training regime. In Frank et al., 2005, training took place after processing a complete sentence, that is, weights were only adjusted after processing the last word. According to the backpropagation algorithm, the change in weight of the connection from unit i to unit j depends on the activation

Table 3. Sentences in the two test sets ('old': sentences describing situations that were also present in the training set; 'new': sentences describing situations that were not in the training set).

set	sentence
old	*bob and jilly plays soccer*
	bob and jilly plays soccer outside
	bob and jilly plays soccer inside
	bob and jilly plays soccer and is tired
	bob and jilly plays soccer and is awake
	bob and jilly plays soccer and wins
	bob and jilly plays soccer and loses
	jilly and bob plays a_computer_game
	jilly and bob plays a_computer_game outside
	jilly and bob plays a_computer_game inside
	jilly and bob plays a_computer_game and is tired
	jilly and bob plays a_computer_game and is awake
	jilly and bob plays a_computer_game and wins
	jilly and bob plays a_computer_game and loses
	bob is tired and plays soccer
	bob is outside and tired
	bob plays hide_and_seek and is awake
	bob is awake and wins
	jilly plays a_computer_game and is tired
	jilly is tired and inside
	jilly is awake and plays with_the_dog
	jilly loses and is awake
new	*bob plays with_the_dog inside*
	bob is inside and plays with_the_dog
	jilly plays with_the_dog inside
	jilly is inside and plays with_the_dog
	bob and jilly plays with_the_dog inside
	bob and jilly is inside and plays with_the_dog
	jilly and bob plays with_the_dog inside
	jilly and bob is inside and plays with_the_dog
	bob plays hide_and_seek outside
	bob is outside and plays hide_and_seek
	jilly plays hide_and_seek outside
	jilly is outside and plays hide_and_seek
	bob and jilly plays hide_and_seek outside
	bob and jilly is outside and plays hide_and_seek
	jilly and bob plays hide_and_seek outside
	jilly and bob is outside and plays hide_and_seek

of i: If unit i is not active, the weight does not change. Adjusting the weights only after processing the last word therefore has the result that connection weights from input units are only adjusted if these input units correspond to words that occur at the end of a sentence. The connections from other input units (i.e., *bob*, *jilly*, *and*, *plays*, *is*) keep their initial, random weight.

Clearly, it is better to train the network after every word, which is the method that was used here. Also, the network was trained for a longer time than the Frank et al., 2005, network: Training lasted for 500 instead of 220 cycles. In each of these cycles, the 290 training sentences were presented in random order. In the first 200 cycles, the learning rate parameter was set to a value of $\eta = .05$; in the next 200 cycles, $\eta = .01$; in the final 100 cycles, $\eta = .001$. Ten networks were trained, differing only in their initial weights, which were chosen randomly from a uniform distribution ranging from -0.5 to $+0.5$.

3.3. Comprehension scores

After training the network, the quality of the output is measured by the *comprehension score*, which expresses to what extent the processing of a sentence that describes situation p actually led to belief in the occurrence of p. This value is computed using belief values:

$$\text{compr}(p) = \frac{\tau(p|X(p)) - \tau(p)}{\tau(p|p) - \tau(p)},$$

where $X(p)$ is the network's output to a sentence describing p.

Before training, $X(p)$ is just a random vector. The expected belief value of p in the random situation represented by $X(p)$, that is, $\tau(p|X(p))$, therefore equals p's a priori belief value $\tau(p)$, resulting in a comprehension score of $\text{compr}(p) = 0$. A positive comprehension score results when $\tau(p|X(p)) > \tau(p)$, that is, the output represents a situation in which p is more likely to occur than a priori. Likewise, the comprehension score is negative when the output represents a situation in which p is *less* likely to occur than a priori. This indicates the network misunderstood the sentence. Ideally, the output to the sentence is identical to $\mu(p)$, the situation vector representing p, which yields $\tau(p|X(p)) = \tau(p|p)$ so $\text{compr}(p) = 1$.

Special care needs to be taken for sentences consisting of two clauses connected by *and* (e.g., *Jilly is outside and tired*) because in these cases the comprehension score may be misleading. If only one half of the statement (e.g., *Jilly is outside*) has been properly understood, the output will be similar to $\mu(\text{J outside})$. As a result, the comprehension score for 'J outside $\wedge$ J tired' will be positive, even though 'J tired' is not represented in the output. For this reason, comprehension scores of sentences consisting of two clauses are also computed for the two clauses separately.

4. RESULTS

4.1. Comprehension scores

As is clear from Table 4, the comprehension scores are significantly above 0 for sentences in the training set and for sentences in both test sets. This is true for the complete sentence as well as individually for the first and second clause of a conjunction. In contrast, Frank et al., 2005, found negative average comprehension scores for the first clause of 'new situation' test sentences.

4.2. Error percentages

Table 5 shows the percentage of sentences that were misunderstood, that is, whose comprehension score was negative. Only the first statement of test sentences describing new situations are misunderstood quite often. However, this percentage is much lower than the error rate of almost 70% found by Frank et al., 2005, for these sentences.

Table 4. Amounts of comprehension, averaged over n values, and 95% confidence intervals, for training sentences, test sentences describing old situations, and test sentences describing new situations.

		clause		
set	n	both	first	second
training	2900	.27 ± .01	.27 ± .01	.39 ± .01
test – old	220	.24 ± .03	.26 ± .04	.33 ± .05
test – new	160	.18 ± .02	.27 ± .04	.39 ± .04

Table 5. Percentages of misunderstood sentences, for training sentences, test sentences describing old situations, and test sentences describing new situations.

	clause		
set	both	first	second
training	7.1%	8.4%	3.1%
test – old	10.9%	8.6%	8.2%
test – new	0%	20.6%	4.4%

5. CONCLUSION

As was argued in the Introduction, a model of sentence comprehension should show two kinds of generalization: It should be able to successfully process new descriptions of the situations it was trained on, and it should be able to comprehend descriptions of situations it was not trained on. In the current implementation, this means that the network should have learned that *and* is commutative and how the words *inside* and *outside* affect the situation being described. The network did manage to learn both these things, as is apparent from the positive comprehension scores (and low error rates) on the 'old situations' and 'new situations' training sets, respectively. These results are much better than those of Frank et al., 2005, who found *negative* average comprehension scores and error rates of well above 50% for the first statement of sentences in the 'new situations' test set.

The network did not simply learn an association between words (or word sequences) and situations, but it has learned how *combinations* of words refer to *combinations* of situations. Naturally, it cannot yet be concluded that simple recurrent networks are able to learn this kind of compositionality in general. First, the model must be applied to a more complex language describing a more complex world.

Also, a more realistic microlanguage and microworld are needed before the network can be taken as a serious candidate for a sentence comprehension model. Of course, successfully teaching a neural network to comprehend even realistic sentences does not automatically make it a model of sentence comprehension: Any cognitive model should also account for empirical data. No comparison between the model's result and experimental data was presented here, but Frank et al., 2005, show that the representation of sentences that develops in the intermediate layer can account for experimental findings by Fletcher and Chrysler, 1990. These researchers found that two sentences are more easily confused in a sentence recognition task if they differ at fewer levels of representation. That is, two different sentences that describe the same proposition (and, therefore, the same situation) are more often confused than two sentences that describe different propositions but the same situation. These latter sentences, in turn, are harder to tell apart than two sentences describing different situations.

During training, the network develops an internal representations on its intermediate layer. After processing a sentence, the activation vector of this layer can be taken as a representation of the sentence. Frank et al., 2005, found that the euclidean distance between two such sentence vectors was related to the results of Fletcher and Chrysler, 1990: Two sentences that differ only textually have vectors that lie closer together than two sentences that differ propositionally but not situationally, while two sentences that do differ situationally have vectors that lie even further apart. If the distance between two sentence vectors is taken to be a measure of the ease with which the sentences can be told apart, this accounts for Fletcher and Chrysler's confusability findings.

The current model can only account for a very limited amount of data. This is largely the result of its lack of a notion of processing time: Processing each word always takes one sweep of activation through the network. Therefore, there is no correlate of word or sentence reading times, which are often measured in experiments on sentence comprehension. If the model is to account for reading times, it needs to be changed into a dynamical system. As argued by van Gelder, 1998, psychological processes cannot be modeled without including a notion of the time in which these processes take place. Therefore, cognitive models should be dynamical systems that can be defined by differential equations to which processing time is inherent. Since the current network is not a dynamical system, it should not even be considered a cognitive model by those standards.

Turning the non-dynamical recurrent network into a dynamical one is actually not hard to do. All that is needed are top-down connections from the output layer to the hidden layer. With this addition, processing each word would cause activation to bounce up and down between these two layers, taking a varying amount of processing time to stabilize. The time required for the network to settle can then be taken as a measure of word reading time. Also, varying the criterion for stabilization may result in an account of processing depth. Frank et al., 2003, successfully modeled the effect of depth-of-processing in this manner. Moreover, including top-down connections is realistic regarding the growing empirical evidence that sentence processing does involves top-down influence from world knowledge and from the events described in the discourse context (e.g., Grodner et al., 2005; Spivey-Knowlton and Sedivy, 1995; van Berkum et al., 2003). Unfortunately, training such a dynamical network still poses quite a technical challenge. Attempts to use the recurrent backpropagation algorithm by Pineda, 1987, or a continuous-time version of the algorithm by Williams and Zipser, 1989, have been unsuccessful so far.

References

Budiu, R. and Anderson, J. R. (2004). Interpretation-based processing: a unified theory of semantic sentence comprehension. *Cognitive Science*, 28:1–44.

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

Fletcher, C. R. (1994). Levels of representation in memory for discourse. In Gernsbacher, M., editor, *Handbook of psycholinguistics*, pages 589–607. San Diego, CA: Academic Press.

Fletcher, C. R. and Chrysler, S. T. (1990). Surface forms, textbases, and situation models: recognition memory for three types of textual information. *Discourse Processes*, 13:175–190.

Frank, S. L., Koppen, M., Noordman, L. G. M., and Vonk, W. (2003). Modeling knowledge-based inferences in story comprehension. *Cognitive Science*, 27:875–910.

Frank, S. L., Koppen, M., Noordman, L. G. M., and Vonk, W. (2005). Modeling multiple levels of text representation. In Schmalhofer, F. and Perfetti, C., editors, *Higher level language processes in the brain: inference and comprehension processes*. Mahwah, NJ: Erlbaum.

Grodner, D., Gibson, E., and Watson, D. (2005). The influence of contextual contrast on syntactic processing: evidence for strong-interaction in sentence comprehension. *Cognition*, 95:275–296.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, UK: Cambridge University Press.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95:163–182.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59:2229–2232.

Rohde, D. L. T. (2002). *A connectionist model of sentence comprehension and production*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. and McClelland, J., editors, *Parallel distributed processing. Volume 1: Foundations*, pages 318–362. Cambridge, MA: MIT Press.

Spivey-Knowlton, M. J. and Sedivy, J. (1995). Resolving attachment ambiguities with multiple constraints. *Cognition*, 55:227–267.

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

van Berkum, J. J. A., Brown, C. M., Hagoort, P., and Zwitserlood, P. (2003). Event-related brain potentials reflect discourse-referential ambiguity in spoken language comprehension. *Psychophysiology*, 40:235–248.

van Dijk, T. A. and Kintsch, W. (1983). *Strategies of discourse comprehension*. New York: Academic Press.

van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21:615–665.

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