Computational modeling
of
discourse comprehension

Cover illustration: *Jip en Janneke voetballen met de situatieruimte* [Bob and Jilly play soccer with situation space] by the author, after a drawing by Fiep Westendorp.

Printed and bound by Ponsen & Looijen b.v., Wageningen.
ISBN 90-74029-26-4

Computational modeling
of
discourse comprehension

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Tilburg,
op gezag van de rector magnificus,
prof. dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen
ten overstaan van een
door het college voor promoties
aangewezen commissie
in de aula van de Universiteit
op vrijdag 27 februari 2004
om 14.15 uur

door

Stefan Lennart Frank
geboren op 11 juni 1973 te Sassenheim

Promotores: prof. dr. L.G.M. Noordman
prof. dr. W. Vonk

Copromotor: dr. M. Koppen

The research presented in this thesis was supported by grant 575-21-007 of the Netherlands Organization for Scientific Research (NWO).

No human inquiry can be a science unless it pursues its path
through mathematical exposition and demonstration

Leonardo da Vinci

Contents

1
Introduction

Sentences rarely occur in isolation. Usually, a sentence forms part of a discourse and cannot be fully understood without relating it to the other discourse statements. As an example, the story *A Snowman with a Broom* by Annie M.G. Schmidt (1963), meant for children aged five years and up, begins with the following six sentences:

> Father, how do you make a snowman? Bob asks. I will help you, father says. He takes a shovel from the shed. And Bob gets a small shovel. And so does Jilly. And then they work very hard. (p. 46. Translated from Dutch)

Although the story is very easy to understand, making sense of any individual sentence is impossible without using information from the rest of the text and, often, the reader's or listener's general knowledge. Under normal conditions, this part of language comprehension proceeds automatically and without much effort. For instance, without being mentioned explicitly it is immediately understood from the above story fragment that

- Father says that *father* will help *Bob make a snowman*.
- *Father* takes a shovel from the shed *in order to help Bob make a snowman*.
- Bob gets a small shovel *from his father*.
- Jilly *gets a small shovel from Bob's father*.
- *Bob, Jilly, and Bob's father* work very hard *on making a snowman*.

This list is far from exhaustive. For instance, it does not say anything about the order in which all these events occur, about the fact that there must be snow around, or about the likelihood that the snowman will get a carrot for a nose. Many of such inferences need to be drawn in order to comprehend the story, and most readers and listeners do understand simple stories without engaging in any conscious problem solving. This cognitive process of discourse comprehension forms the topic of this thesis. In particular, the thesis deals with computational models, that is, theories of discourse comprehension that are described precisely and completely enough to be formalized mathematically and implemented as computer programs. First, the remainder of this chapter introduces some of the relevant issues in discourse comprehension and computational modeling.

1.1 Discourse comprehension

Three central questions in the study of discourse comprehension are what units of meaning make up a discourse, how discourse is represented mentally, and how information is inferred from a discourse. Here, each of these issues is discussed briefly.

1.1.1 Propositions

The smallest unit of meaning usually identified in the study of discourse is the proposition. There are two views on the nature of propositions and although these do not exclude each other, much confusion can be avoided by distinguishing between them. First, a proposition can be regarded as a statement to which a truth value can be assigned. For instance, the sentence *The cat is on the mat* corresponds to a proposition that is true if and only if a particular cat actually is on a particular mat.

The words *cat*, *on*, and *mat* individually are not propositions because nothing about them can be 'true' or 'false'. The relation among the three concepts CAT, ON, and MAT,[1] however, does constitute a proposition. This is the second, structural, view on propositions. Indicating that they can be thought of as relations between concepts, propositions are often denoted in the form PREDICATE(ARGUMENT1,ARGUMENT2,...), where PREDICATE denotes the nature of the relation between the ARGUMENTS, the number of which can vary. The roles of the different arguments are indicated by their order, but prepositions may be added to avoid confusion. Arguments can also be propositions themselves. For instance, the sentence *Father says that he will help Bob make a snowman* would correspond to the proposition SAYS(FATHER,HELP(FATHER,BOB,MAKE(BOB,SNOWMAN))), which has as second argument the proposition HELP(...,...,...), embedded in which is the proposition MAKE(...,...).

Although there have been attempts to construct general guidelines (e.g., Turner & Greene, 1978), extracting the propositional structure from a text remains for a large part a subjective task. For instance, three experts who were asked to propositionalize the simple sentence *Lyle pushed Paris out of his mind*

[1] Throughout this thesis, examples of literal texts are printed in *italic* font, while concepts and propositions are indicated by SMALLCAPS.

for three months did not even agree on the number of propositions it contains. Depending on the expert, the analysis resulted in either three, five, or seven propositions (Perfetti & Britt, 1995, Note 3).

Quite a lot of research has gone into finding out whether propositions form part of a text's mental representation. Goetz, Anderson, and Schallert (1981) found that subjects often recall all of a proposition or none of it. This has often been interpreted as evidence for the cognitive reality of propositions (e.g., Fletcher, 1994; Kintsch, 1998, chap. 3.1; Van Dijk & Kintsch, 1983, chap. 2.2), which demonstrates the confusion that results from not differentiating between the two views on propositions. Although Goetz et al. do show that the units of a text's mental representation may correspond to propositional units, all-or-nothing recall of such units can in fact be interpreted as evidence *against* the existence of propositional *structures*. If subjects never recall part of a proposition, it is very well possible that it does not have any parts. In that case, propositions are represented holistically and not as a collection of related concepts.

Ratcliff and McKoon (1978) performed an experiment designed to show that propositional structures are part of a text's mental representation. They had subjects read sentences such as *The mausoleum that enshrined the tzar overlooked the square*, which consists of two propositions: ENSHRINED(MAUSOLEUM,TZAR) and OVERLOOKED(MAUSOLEUM,SQUARE). If the mental representation of the sentence also contained these propositional structures, so they hypothesized, the words *square* and *mausoleum*, which share a proposition, should prime each other more strongly in a recognition task than the words *square* and *tzar* do, even though the words of this latter pair are closer together in the sentence. Indeed, they did find stronger priming between words that share a proposition than between words that do not, and concluded that propositional structures are cognitively real. However, as is suggested by the above example, they seem not to have taken into account that readers may form a mental image of the events in the text instead of a propositional structure. As noted by Zwaan (1999), the effect on priming might have occurred because, in this mental image, the square and the mausoleum are closer together than the square and the tzar, or even because the tzar, being inside the mausoleum, is not visible from the square.

The same problem occurs in the texts used as experimental stimuli by Dell, McKoon, and Ratcliff (1983), which were taken from McKoon and Ratcliff (1980). One of these reads

> A burglar surveyed the garage set back from the street. Several milk bottles were piled at the curb. The banker and her husband were on vacation. The criminal slipped away from the streetlamp. (Dell, McKoon, & Ratcliff, 1983, Table 1; McKoon & Ratcliff, 1980, Table 1)

After reading the word *criminal* in the last sentence, recognition of *garage* was found to be faster than after reading a similar text in which *criminal* was replaced by *cat*. This effect was explained by assuming a propositional representation. The text's first sentence gives rise to the proposition SURVEYED(BURGLAR, GARAGE). The anaphor *criminal* in the last sentence refers to the burglar and therefore activates BURGLAR in the reader's mental representation. This results in activation of the concept GARAGE because BURGLAR and GARAGE share the proposition coming from the first sentence.

Such within-proposition activation between concepts can be taken as evidence that the story's mental representation does consist of propositional structures. As in Ratcliff and McKoon's (1978) experiment, however, the stimuli do not seem to have been controlled for the mental image they might evoke in a reader. Experimental findings by Zwaan, Stanfield, and Yaxley (2002) support the hypothesis that the reader of a text obtains a mental image of the scene described by the text. Concepts from the same proposition tend to be close together physically in this scene. In the above example, the burglar is probably very close to the garage in order to survey it. Therefore, focusing attention to the burglar in the mental image of this scene will also highlight the garage.

To conclude, although propositional structures are often assumed, their status in the human cognitive system is not that well established.

1.1.2 Levels of representation

Ever since this was proposed by Kintsch and Van Dijk (1978; see also Van Dijk & Kintsch, 1983), the mental representation of discourse has been assumed to involve three distinct levels. The first level is the *surface representation*, consisting of the text's literal wording. This representation is quite short-lived: The literal text is usually forgotten quickly.

The surface representation gives rise to the second level, called the *textbase*, where the meaning of the text is represented. Commonly, this meaning is expressed in propositional units although Kintsch (1988, 1998) also includes concepts in the textbase. For the textbase, only propositional structure is relevant.

Two propositions in the textbase that share at least one of their arguments are considered connected, resulting in a network of propositions. If a proposition is read for which no argument-sharing proposition can be found, the relation between the current proposition and the rest of the textbase needs to be inferred somehow. At this point, the discourse representation obtains information that is not literally present in the statements of the discourse. Kintsch and Van Dijk do not explain how these inferences come about, but they do note that

> most of the inferences that occur during comprehension probably derive from the organization of the text base into facts that are matched up with knowledge frames stored in long-term memory, thus providing information missing in the text base by a process of pattern completion. (Kintsch & Van Dijk, 1978, p. 391)

These "facts" refer to the reader's "personal interpretation of the text that is related to other information held in long-term memory" (Kintsch, 1998, p. 49). This so-called *situation model* (Kintsch, 1998; Van Dijk & Kintsch, 1983) forms the third level of text representation, which is where most knowledge-based inferences are represented. Situation models are "integrated mental representations of a described state of affairs" (Zwaan & Radvansky, 1998, Abstract). They can be thought of as similar to the representation that results from directly experiencing the events described in the text (Fletcher, 1994). Unlike the textbase, the situation model is not concerned with structural relations among propositions, such as argument overlap. Instead, propositions in the situation model are related by the effects they have on one another's truth values: "relations between facts in some possible world ... are typically of a conditional nature, where the conditional relation may range from possibility, compatibility, or enablement via probability to various kinds of necessity" (Kintsch & Van Dijk, 1978, p. 390).

Several researchers have attempted to show that Kintsch and Van Dijk's three levels are present in the mental representation of discourse (e.g. Kintsch, Welsch, Schmalhofer, & Zimny, 1990). Compelling evidence comes from a series of experiments by Fletcher and Chrysler (1990). They had subjects read short stories, each describing a linear ordering among five objects. For instance, one of the stories read

> George likes to flaunt his wealth by purchasing rare art treasures. He has a Persian rug worth as much as my car and it's the cheapest thing he owns. Last week he bought a French oil painting for $12,000 and an

> Indian necklace for $13,500. George says his wife was angry when she found out that the necklace cost more than the carpet. His most expensive "treasures" are a Ming vase and a Greek statue. The statue is the only thing he ever spent more than $50,000 for. It's hard to believe that the statue cost George more than five times what he paid for the beautiful Persian carpet. (Fletcher & Chrysler, 1990, Table 1)

In this example, five art treasures can be ordered by price: rug/carpet, painting, necklace, vase, and statue. After reading ten of such stories, subjects were given from each story one sentence without its final word. Their task was to choose which of two words was the last of the sentence. For the story above, the test sentence was *George says his wife was angry when she found out that the necklace cost more than the …* and subjects might have to recognize either *carpet* or *rug* as the actual last word of this sentence in the story they read. Since *carpet* and *rug* are synonyms, the difference between them appears at the surface text level only. If subjects score better than chance on this decision, they must have had some kind of mental representation of the surface text.

Alternatively, the choice might be between *carpet* and *painting*. Since these are not synonyms, this comes down to a choice between different propositions: One states the necklace costs more than the carpet, while according to the other the necklace costs more than the painting. Scoring better on this choice than on the choice between *carpet* and *rug* shows the existence of a level of representation beyond the surface text.

In fact, the necklace cost more than both the carpet and the painting. Subjects who erroneously choose *painting* over *carpet* do not violate the situation model since their choice will still result in a statement that is true in the story. However, if the choice is between *carpet* and *vase*, different choices correspond to different situation models. If subjects score better on this choice than on the choice between *carpet* and *painting*, they must have developed a situation-level representation.

Indeed, Fletcher and Chrysler (1990) did find a better than chance score on the choice between synonyms, an even higher score on the choice between propositions, and the highest score on the choice between situation models. This result strongly supports the existence of at least three levels of representation. What remains unclear, however, is how these representations are constructed during sentence and discourse comprehension. Somehow, the surface text should be parsed into propositions that form a textbase. Next, items from this textbase should activate relevant parts of the reader's world knowledge, resulting in a situational representation including inferred facts. How these two processes operate is still an open question.

1.1.3 Coherence and inference

As is clear from the story fragment at the beginning of this chapter, it is almost impossible for a text to provide a full situation model. Even the textbase may not be completely specified. For instance, when Bob asks his father how to make a snowman, the two pronouns in father's answer *I will help you* need to be resolved to find the arguments of the proposition HELP(FATHER,BOB).

Discourse statements are interrelated and part of their interpretation depends on the relations among them. When information is lacking from the text some of it can, and often needs to, be inferred in order to achieve sufficient comprehension. Possible inferences range from finding the correct referent of a pronoun to inferring details of the state of affairs at any moment in the story, but only few of these inferences are actually made during reading (for an overview, see Garrod & Sanford, 1994; Singer, 1994; Van den Broek, 1994).

There has been considerable debate on which inferences are made during reading. According to McKoon and Ratcliff's (1992) minimalist hypothesis, inferences are made to obtain *local coherence*, that is, each discourse statement is related to the one or two statements immediately preceding it. Apart from these inferences, only "those based on easily available information" (p. 441) are made. Information can be easily available because it is explicitly mentioned in the text or because it follows from "well-known general knowledge" (p. 441). However, as Noordman and Vonk (1998) point out, this hypothesis lacks an independent criterium to determine which general knowledge is well-known and which is not.

In contrast to the minimalist hypothesis, the constructionist hypothesis (Graesser, Singer, & Trabasso, 1994) claims that, during normal reading, a reader tries to explain the events described in the text. For this to be successful, local coherence is not always sufficient. Therefore, there is also an effort to establish *global coherence*, meaning that the current statement is related to the entire preceding text and not only to just a few immediately preceding statements.

The problem with such hypotheses is that they assume standards for 'normal reading' or 'sufficient comprehension', while these depend on the characteristics and goals of the individual reader. Noordman and Vonk (1992) showed that logically inferable facts required for local coherence are inferred from an expository text only by readers who already know these facts. However, readers who do not know them do infer the facts if it is useful for their reading purpose, for instance because they have to answer specific questions or check the text for inconsistencies (Noordman, Vonk, & Kempff, 1992). According to Keenan, Potts, Golding, and Jennings (1990), the experimental method that is used strongly affects whether or not an inference is detected. This makes it even less clear what types of inferences are drawn under which conditions.

In general, it may be impossible to make strong predictions concerning the inferences that will or will not be drawn. All that can be concluded is that inferences are more likely to be made, or are made to a greater extent, when they are more important to the reader's goals, require knowledge that is easily available, and contribute to the coherence of the text (Noordman et al., 1992; Vonk & Noordman, 1990). Noordman and Vonk (1998) argue that it is more important to develop a theory of the underlying process of inferencing than a theory that merely states which inferences will be drawn under which conditions. A useful theory of the process of knowledge activation by incoming text and elaboration of the text's mental representation would make predictions regarding the inferences made during this process. The research presented in this thesis is aimed at the development of just such a theory.

1.2 Computational modeling

In order to explain experimental findings in psychology, models of the underlying process are constructed. Until recently, such models were mainly expressed verbally, that is, without requiring equations or completely specified algorithms. As an example, the first rule of the constructionist theory of discourse comprehension (Graesser et al., 1994) states that if the statement being read describes a character's intentional action or goal, the reader searches his or her working memory and long-term memory to find a superordinate goal of the stated action or goal. In combination with the theory's other rules it constitutes a verbal model that is quite complex, but not too complex to ascertain its internal consistency or make qualitative predictions. As a model's complexity increases, however, it becomes more difficult to test without actually implementing and running it as a computer program. Dijkstra and De Smedt (1996) mention several more reasons for engaging in computational modeling: It can support the interpretation of empirical results, suggest new experiments, or even simulate experiments that cannot be performed in practice.

Turning a verbal model into a computational one involves precisely formalizing many aspects that can stay vague in the verbal model. In the example above, it must be specified how *exactly* it is determined whether a discourse statement is an intentional action or a goal, how memory is searched, and how a superordinate goal is recognized. Probably most important of all, however, is to specify the representation of the discourse and the reader's knowledge, since these representations are needed before any processes operating on them can be implemented.

1.2.1 Representation

Following Kintsch and Van Dijk's (1978) idea that discourse can be represented at the textbase level as a network of connected propositions, most psychologically motivated computational models of discourse comprehension are so-called connectionist models.[2] Such models consist of a large number of simple processing elements that form nodes in a network. The nodes can become

[2] In linguistics, the standard model of discourse representation is Kamp's (1981) Discourse Representation Theory. This theory and related approaches, which are rooted in logic and formal

'activated' and influence the activation of nodes they are connected to. These connections can encode properties of the discourse (e.g., in the Construction-Integration model; Kintsch, 1988), the reader's memory trace (e.g., in the Landscape model; Van den Broek, Risden, Fletcher, & Thurlow, 1996), or the reader's world knowledge (e.g., in the model by Golden & Rumelhart, 1993). Alternatively, it may not be possible to assign a meaningful psychological or textual label to individual connections (e.g., in the Story Gestalt model; St. John, 1992).

Whether or not meaningful labels can be assigned to the model's processing elements defines the distinction between *localist* and *distributed* representations, which shall be one of the main issues in this thesis. In a localist representation, there is a one-to-one mapping between the model's processing elements and the represented objects (e.g., concepts or propositions). Each element corresponds to one object, and each object is represented by one element. The main advantage of such a representation lies in its simplicity. Building a localist representation is relatively easy, and interpreting the model's output is straightforward.

If a representation is distributed, there is no one-to-one mapping between the processing elements and the represented objects. Instead, a pattern of activation over all processing elements forms a representation. Distributed representations are much harder to develop than localist ones. However, considering their advantages (see e.g. Hinton, McClelland, & Rumelhart, 1986) using distributed representations may be worthwhile.

For modeling discourse comprehension, the most important of these advantages may be the way new objects can be represented. Since new concepts and propositions can be constructed from known ones, it is not possible to define in advance everything that may need to be represented in a discourse comprehension model. For such a model, therefore, one particularly useful feature of distributed representations is their ability to easily encode novel objects. If a new object needs to be represented in a localist model, the model needs an extra processing element. For most of the localist models discussed in this thesis, this means that a discourse is represented as a growing network. Every time a new discourse statement is processed, one or more nodes representing the statement need to be added to the network, and the relations to the previous discourse (i.e., the connections to the rest of the network) need to be determined.

linguistics instead of psychology and do not include a psychologically motivated process, shall not be discussed in this thesis.

In a distributed model, new objects can be represented more elegantly. A pattern of activation representing the new object needs to be chosen, but the number of processing elements can stay the same. Since new concepts and propositions usually are related somehow to the concepts and propositions from which they are constructed, the new representation can be chosen on the basis of this relationship.

1.2.2 Model evaluation criteria

There exist many different models in cognitive psychology, but no standards for quality determination which are agreed upon and can be applied objectively. Nevertheless, Jacobs and Grainger (1994) do list several criteria for the evaluation of models. In particular, they note that good models should be simple, descriptively adequate, explanatorily adequate, and general.

Simplicity
Although it seems intuitively clear what is meant by simplicity, it is a very hard notion to define or determine. The clearest measure of simplicity is the number of free parameters in the model. Having fewer parameters generally means a simpler model, but this does not need to be true in general since the meaning of the parameters should also be taken into account. A set of parameters that can be interpreted as psychological measures (e.g., working memory size) may be preferred to a smaller set of parameters that do not mean anything but simply do the job.

Jacobs and Grainger (1994, p. 1317) claim that "the number and length of equations . . . are straightforward measures of simplicity". Although there may be some truth in this, it should also be considered how the model's equations arise. If a single, short equation is an ad hoc construction that may as well have been different, it does not constitute a simple model. If, on the other hand, the equations follow mathematically from simple assumptions on which the model is based, it does not matter how many are needed to express the model, nor how long they are.

Descriptive adequacy
Descriptive adequacy refers to the ultimate test for any cognitive model: its ability to predict experimental data. The more data is accounted for, the more

descriptively adequate the model is. There is, however, a trade-off with simplicity. In theory, any data can be produced by some set of equations and parameter settings, but such a set hardly constitutes a model if it is not constrained to show at least some simplicity.

When having to choose between simplicity and descriptive adequacy, Dirac (1963, p. 47) claims that "it is more important to have beauty in ones equations than to have them fit experiment". This may be true in particular when dealing with models of discourse comprehension since this process involves far more factors than can ever be implemented in any model, making extreme simplification unavoidable. For instance, understanding a text usually requires the reader to apply his or her knowledge, but no realistic amount of such knowledge can be made available to a model. Also, the input to a model is usually a pre-parsed version of the stimuli used in experiments, if there is any relation between the two at all. As a result, precise predictions of experimental data (i.e., a quantitative fit between the data and the model's results) cannot be expected. Since only a qualitative fit (i.e., a comparison between data and results on an ordinal scale) is possible, adding parameters and equations just to achieve a quantitative fit does not result in a better model.

Explanatory adequacy

A simple model that predicts empirical data can be considered a good model, but this does not necessarily make it useful. One of the reasons to engage in computational modeling in the first place is to explain some cognitive phenomenon. Often, models are specifically designed to produce certain empirical data, and it is doubtful to what extent such a model can be said to *explain* these data. If, however, the model also shows a desired effect that it was *not* designed to show, it does give an explanation for that effect. A model without such emergent properties has less explanatory adequacy.

Generality

The more widely a model can be applied, the higher its generality. Jacobs and Grainger distinguish several types of generality but this thesis shall only deal with *stimulus generality* and *task generality*. The first refers to the model's ability to process different inputs. A discourse comprehension model whose design or parameters have to be adjusted for the specific discourse that is to be comprehended is less general than a model that can readily process different stimuli.

The second type of generality refers to the cognitive processes the model can simulate. Of course, the more tasks a model can perform, the higher it scores on tasks generality.

A third type of generality, *response generality*, is concerned with the empirical measures the model can be validated against. A model whose output can be related to, for instance, reading times, error rates, and recall data, has higher response generality than a model that produces only one of these measures. Since it is a necessary condition for descriptive adequacy, we shall not investigate response generality independently.

Thesis overview

In the next chapter, seven discourse comprehension models from the literature are presented and evaluated critically. Following this, an eighth model is discussed in a separate chapter. This particular model requires special scrutiny since it shares its architecture with the Distributed Situation Space model of knowledge-based inferencing, presented in Chapter 4, which forms the central part of this thesis. By adding three extensions to the model, Chapter 5 shows how the model can be applied to tasks beyond those it was originally designed for. The final chapter summarizes our findings, claims, and conclusions.

2 Models of discourse comprehension

A paper based on this chapter is to be submitted for publication (Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W., 2004, *[Discourse comprehension models: a critical analysis]*, manuscript in preparation).

This chapter discusses seven models of discourse comprehension: the Resonance model, the Landscape model, the Langston and Trabasso model, the Construction-Integration model, the Predication model, the Sentence Gestalt model, and the Story Gestalt model. The focus of these models varies strongly, from the short-term fluctuation of activations of discourse items (Resonance) to the causality based, long-term memory representation of the discourse (Langston and Trabasso). As a result, a direct comparison among the different models is impossible. Instead, qualities and limitations are discussed for each model individually.

Focus will lie mainly on computational and mathematical issues. In spite of the differences among the models, computational similarities can often be identified. In order to make it easier to compare the models' computational descriptions, and to avoid confusion, we have tried to apply one standardized notation for all models as much as possible. The notation used in this chapter can therefore differ from those in the models' original presentations.

Most models consist of a number of processing elements. In localist models, each element corresponds to a meaningful unit such as a concept, a proposition, or another item from the text or the reader's knowledge. Such items, as well as the model's corresponding processing elements, shall be denoted by the symbols $p, q, r, \ldots$. A processing element p has at least one variable value associated with it, which is denoted by x_p. The collection of values for all elements forms the row vector $X = (x_p, x_q, \ldots)$. Often, there also exists a value w_{pq} associated to any *pair* of processing elements (p, q). This value is not necessarily the same as w_{qp}, the value associated to the reversed pair (q, p). The collection of all ws forms the matrix W.

Usually, the elements' values fluctuate as the model goes through a number of processing cycles. These cycles are indexed by the symbol c, and the vector of values in cycle c is denoted by $X(c)$. Initially, $c = 0$ so $X(0)$ denotes the model's initial state. If there exists a parameter that controls the moment at which the process halts, it is denoted by θ.

If a model processes a sequence of text sentences or narrative events, these are indexed by the symbol t. If needed, it is added as a subscript to X, so $x_{p,t}(c)$ denotes the value of processing element p in processing cycle number c during processing of sentence/event number t.

2.1 The Resonance model

As reading proceeds, some parts of the reader's mental representation of the text are more accessible than others. For instance, concepts and propositions that are central to the text can remain in working memory while less important elements are backgrounded. However, previously backgrounded text items can become reactivated if this is required or instigated by the sentence currently being read. This phenomenon is known as *reinstatement*.

Basically, there are two explanations for reinstatement: top-down and bottom-up. The top-down interpretation states that readers actively try to link incoming text statements to earlier ones. If a link cannot be made with the current contents of working memory, the mental representation of the text may be searched until a connection can be made. This causes earlier text elements to be reinstated into the reader's working memory. Alternatively, the bottom-up interpretation claims that there is no active search process. Instead, elements from the current sentence automatically activate previous statements in which similar elements occurred, reinstating them into working memory.

Albrecht and Myers (1995) conducted an experiment from which they concluded that reinstatement is a bottom-up process. They claim that elements from the reader's mental representation of the text can *resonate* to the elements in the sentence being processed. The Resonance model (Myers & O'Brien, 1998) is a formal description of this bottom-up reinstatement process. Since the model was designed specifically to explain the results of Albrecht and Myers' experiment, we begin with a discussion of that experiment.

2.1.1 The captain's inventory

Table 2.1 shows one of the texts used by Albrecht and Myers (1995). This 'captain text' can be divided into three episodes. In the first, a captain is introduced who is sitting at his desk because he has to finish his ship's inventory before his shore leave can begin. However, before he starts the inventory, he is called away for urgent captain-business. The second episode does not include any reference to the unfinished inventory. As a result, the information about the inventory is assumed no longer to be present in the reader's working memory. In the third episode, the captain returns to his office, sits at his desk, and is

Table 2.1: The captain text (Myers & O'Brien, 1998, Table 1).

t	sentence
1	The cruise was coming to an end and the ship would soon dock.
2	The captain sat in his office, trying frantically to finish some paperwork.
3	He had to do an inventory of the ship before he could begin his leave.
4	He had been heavily fined for not completing the inventory on an earlier cruise.
5	He pulled up his chair and sat down at his large desk.
6	However, before he could start the inventory, some passengers arrived to report a theft.
7	He would have to complete the inventory later.
8	He left his desk covered with the inventory forms and began an investigation in order to catch the thief.
9	He carefully reviewed each of the complaints.
10	After a few minutes, he was sure the thief was a staff member.
11	It was someone who had access to a master key to the passengers' cabins.
12	This greatly reduced the number of suspects.
13	After questioning a few of the crew members, he was sure the thief was the ship's purser.
14	Within minutes, the purser was locked up.
15	The captain returned to his office and sat down at his large desk.
16	He was happy to be done with the cruise.
17	He was ready to start his shore leave.

claimed to be ready to start his shore leave. This last statement is of course inconsistent with the earlier information that he has to finish the inventory first. The question Albrecht and Myers asked was: Do readers notice this inconsistency?

In order to investigate this, they constructed an alternative text in which the captain *did* finish the inventory in the story's first episode, while the other two episodes were not altered. As a result, sentences 16 and 17 are not inconsistent in the alternative text even though they are identical to sentences 16 and 17 in the original text of Table 2.1. Albrecht and Myers found that subjects took more time to read these two sentences in the original, inconsistent version of the story than in the alternative, consistent version. It was concluded that readers do notice the inconsistency. This means that the information about the unfinished inventory, which was supposedly backgrounded during reading of the story's second episode, must have been reinstated after reading sentence 15. The inconsistency could not have been noticed otherwise.

The next question was whether this reinstatement was the result of a top-down process in which readers try to understand why the captain returns to his desk, or of a bottom-up process in which the words *large desk* of sentence 15 automatically activate the concept INVENTORY because the eighth sentence states that the inventory forms covered the desk. To test this, Albrecht and Myers constructed yet another alternative version of the captain text. In this second alternative, as in the original text, the captain did not finish his inventory so sentences 16 and 17 are inconsistent with the preceding text. However, sentence 15 did not mention the large desk in this alternative text, which means that the inconsistency may not be noticed if reinstatement is a bottom-up process. Indeed, it was found that subjects took less time reading sentences 16 and 17 in this version of the story than in the original version. Apparently, readers did not notice the inconsistency when the large desk was not mentioned in sentence 15. It was concluded that the words *large desk* caused reinstatement of the propositions related to INVENTORY and that, therefore, reinstatement is a bottom-up process.

2.1.2 Model description

The text network

The Resonance model processes the sentences of a text one at a time. However, like most other discourse comprehension models, it cannot process a literal sentence. Each sentence must first be put into an appropriate format, namely a network consisting of items from the sentence. Myers and O'Brien (1998, p. 143) distinguish three types of items: concepts, propositions, and sentence markers. Concepts and propositions form the content of a sentence, and sentence markers act as "local context markers" (p. 143) that group together propositions appearing in the same sentence.

Every time a sentence enters the model, its sentence marker, its propositions, and new concepts from the sentence form nodes that can be connected to each other and to the nodes corresponding to previous text items. Items p and q are connected if one of the following conditions holds (Myers & O'Brien, 1998, pp. 143-144):

- p is a sentence marker and q is a proposition in the corresponding sentence.
- p is a proposition and q is one of its arguments.
- p and q are propositions that have identical arguments.

If none of these applies, p and q are not connected. All connections are symmetrical, so a connection between p and q implies a connection between q and p.

We parsed the first three sentences of the captain text into the 22 items listed in Table 2.2. Figure 2.1 shows the corresponding text network. The full collection of concepts and propositions used in our simulations was based on those provided by J.L. Myers (personal communication to W. Vonk, September 20, 1995) and Weeber (1996) and can be found in Appendix A.1.

Table 2.2: Seven concepts (indicated by C), twelve propositions (P), and three sentence markers (S) corresponding to first three sentences ($t = 1, 2, 3$) of the captain text in Table 2.1.

t	label	meaning
1	S1	(sentence 1 marker)
	C1	CRUISE
	C2	SHIP
	P1	ENDING(CRUISE)
	P2	DOCK(SHIP)
	P3	SOON(P2)
	P4	AND(P1,P3)
2	S2	(sentence 2 marker)
	C3	CAPTAIN
	C4	OFFICE
	C5	PAPERWORK
	P5	SAT(CAPTAIN,in:OFFICE)
	P6	FINISH(CAPTAIN,PAPERWORK)
	P7	TRIES(CAPTAIN,P6)
	P8	FRANTICALLY(P7)
3	S3	(sentence 3 marker)
	C6	INVENTORY
	C7	LEAVE
	P9	OF(INVENTORY,SHIP)
	P10	MUST_DO(CAPTAIN,P9)
	P11	BEGIN(CAPTAIN,LEAVE)
	P12	BEFORE(P10,P11)

Every time the items of a sentence are added to the text network, a connectivity matrix W is constructed. If items p and q are connected, elements w_{pq} and w_{qp} of W receive a value of 1. If p and q are not connected, $w_{pq} = w_{qp} = 0$. The

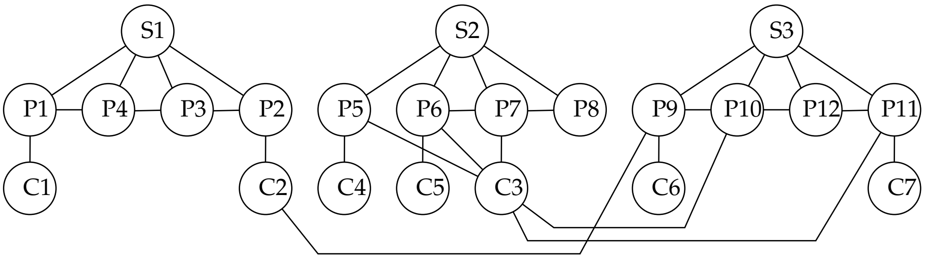

Figure 2.1: The text network after processing the first three sentences of the captain text. The node's labels refer to the text items in Table 2.2.

number of items to which item p is connected is denoted n_p, which equals the sum of the nth column (or, equivalently, the nth row) of matrix W.

The resonance process

After adding the items from the current sentence to the text network, the resonance process described below is executed. This process takes as input the network corresponding to the text read so far and computes a *resonance value* x_p for each item p. All resonance values are initially set to $x_p(0) = 0$. The items that end up with the largest final resonance values are said to remain in working memory after the sentence has been processed.

Apart from a resonance value, to each item p is associated a *signal strength* s_p, which indicates the extent to which p can influence the resonance values of other items. If p is part of the current sentence or remained in working memory after processing the previous sentence, its initial signal strength $s_p(0) = n_p^{-1}$, one divided by the number of connections of the item.[1] Otherwise, $s_p(0) = 0$.

During the resonance process, resonance values and signal strengths are updated over a number of processing cycles. The collection of all resonance values at cycle c forms the resonance vector $X(c) = (x_p(c), x_q(c), \ldots)$. Likewise, the signal strengths form the signal row vector $S(c) = (s_p(c), s_q(c), \ldots)$.

In each processing cycle, items that have a signal strength send a signal to the items they are connected to. As a result, the resonance of a receiving item p increases by the total amount of signal received, which equals $\sum_q s_q(c) w_{pq}$. In

[1] It is unclear whether the arguments of the propositions in the current sentence have an initial signal strength if these arguments already occurred earlier in the text and therefore do not form new nodes in the text network. We obtained better results if these items did have an initial signal strength, so the simulations presented here are based on such an implementation.

more compact vector notation, the resonances in cycle $c+1$ are computed from the resonances and signals in the previous cycle c by

$$X(c+1) = X(c) + S(c)W. \tag{2.1}$$

Next, the signal strengths are updated. An item's signal strength increases as its resonance increases, but decays over processing cycles and is lower for items with a larger number of connected items n_p. Moreover, there exists a threshold parameter θ that controls the level below which the signal strength is set equal to 0. All in all, the signal strength of item p in cycle $c+1$ equals

$$s_p(c+1) = \begin{cases} \frac{1}{n_p}(1-\gamma)^c x_p(c+1) & \text{if } (1-\gamma)^c x_p(c+1) \geq \theta \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where γ is a parameter between 0 and 1, controlling the decay rate of signal strength. Equations 2.1 and 2.2 are iterated until all signal strengths are 0, which always takes a finite number of cycles, as is proven in Appendix B.1. The items that end up with the largest resonance value are said to remain in working memory and receive an initial signal when the next sentence, if any, enters the model.[2] The number of items in working memory is set to four (Myers & O'Brien, 1998, p. 147).

Two notational differences with Myers and O'Brien's description of their model are worth mentioning. First, Myers and O'Brien do not explicitly define the connectivity matrix W. Second, they define a decay parameter β which is related to our γ by $\gamma = 1 - \mathrm{e}^{-\beta}$, resulting in a more complex expression for computing signal strengths. For the low values of β tested by Myers and O'Brien $(0.01, 0.02, \ldots, 0.05)$, their β and our γ are almost equal. For example, Myers and O'Brien (1998, p. 148) found an optimal value of $\beta = 0.02$, which corresponds to $\gamma = .0198$. The levels of the threshold parameter they tested were $\theta = 0.01$ and $\theta = 0.05$, both of which were found to be appropriate. We used a value of $\theta = 0.05$ in our simulations.

[2] It is unclear which items make it to working memory if several have the same large resonance value. When a tie occurred in our simulations, items from more recent sentences were chosen over older ones. Within a sentence, propositions were chosen over concepts, which were chosen over sentence markers.

2.1.3 Evaluation

Amount of reinstatement

The 15th sentence of the captain text is believed to reinstate propositions related to INVENTORY. This concept, the propositions to which it is an argument, and the markers of the sentences that contain them, are called *critical items* (Myers & O'Brien, 1998, p. 147). If the model simulates reinstatement properly, working memory should not contain any critical items after processing sentence 14, but after processing sentence 15 it should. Therefore, the amount of reinstatement can be defined as the number of critical items in working memory after processing the reinstating sentence, minus their number after processing the previous sentence. Since working memory can contain four items, the maximum amount of reinstatement is four. In practice, we found a maximum amount of reinstatement by sentence 15 of two items, for decay rates γ between .021 and .031. The Resonance model does seem to simulate reinstatement. However, this result becomes somewhat less convincing when we take a look at the number of critical elements in working memory after processing other sentences. It turns out that sentence 13, having nothing to do with the inventory or the captain's desk, also brings a critical element into working memory.

To show that reinstatement is a bottom-up process, there should be less reinstatement in an alternative version of the story in which *large desk* is not mentioned in sentence 15. Indeed, Myers and O'Brien (1998, p. 148) report no reinstatement of critical elements by sentence 15 at all when processing this alternative story. We did not find an absence of reinstatement for the alternative story, but there was a decrease from 2 to 1 for decay rates γ ranging from .027 to .031. These values are somewhat different from the optimal decay rate reported by Myers and O'Brien, corresponding to $\gamma = .0198$. This difference may be caused by differences in details of implementation or of the constructed text network.

Recency and connectivity effects

The accessibility of text items depends on more than just reinstatement. Two main findings are that more recently read items are, in general, more available than older items, and that items which are central to the text are more accessible than less important items (Albrecht & Myers, 1991; O'Brien, 1987; O'Brien, Albrecht, Hakala, & Rizzella, 1995). The Resonance model can simulate both these effects. First, items from the current sentence receive an initial signal and can therefore be expected to end up with larger resonance values than other items, resulting in a *recency effect*. Second, items that are more central to the text have many connections to other items and are therefore more likely to receive large resonance values, resulting in a *connectivity effect*. The magnitude of this effect can be defined as the coefficient of determination (r^2) between the numbers of connections of the items $(n_p, n_q, \ldots)$ and their final resonance values $(x_p, x_q, \ldots)$. That is, the magnitude of the connectivity effect is the proportion of variance in resonance values explained by the items' numbers of connections. Likewise, the size of the recency effect can be defined as the proportion of variance in resonance values explained by whether the items occurred in the current sentence or in a previous one.

The value of decay parameter γ can be expected to strongly influence the magnitudes of the recency and connectivity effects. If γ is large, signals decay quickly and resonance will not spread far through the text network, resulting in a strong recency effect. For small values of γ the opposite happens: Signals keep spreading throughout the network and most resonance will eventually settle on the items that have the largest number of connections. Figure 2.2 shows that, in our simulations, a clear trade-off between the recency effect and the connectivity effect, controlled by γ, indeed occurs when processing the captain text.

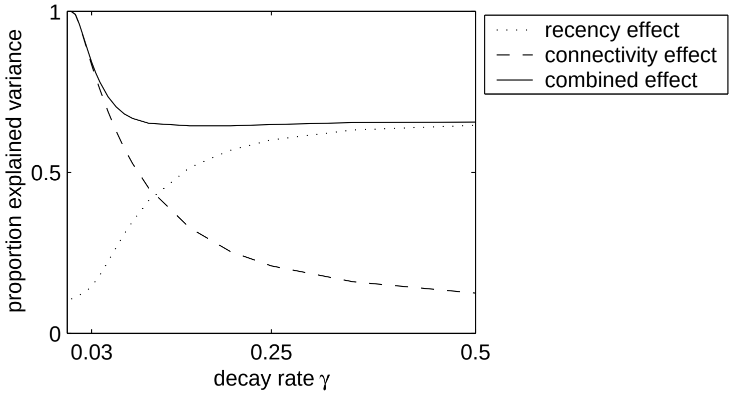

Figure 2.2: Proportion of variance in resonance values explained by recency, connectivity, or both, as a function of decay parameter γ, after processing sentence 15 of the captain text.

The range of decay rates we found to be optimal ($.027 \leq \gamma \leq .031$) results in a very strong connectivity effect and only a small recency effect. It is not surprising that the recency effect must be weak for reinstatement to occur. By definition of reinstatement, critical items are not recent when the reinstating sentence is being processed. Since a weak recency effect means that resonance values of recent (i.e., non-critical) items are low, other items, including the critical ones, have a better chance to make it to working memory when the recency effect is weaker.

Increasing the value of γ strengthens the recency effect, making reinstatement harder. If, on the other hand, the decay rate is lowered, the combined effect of recency and connectivity can become so strong that reinstatement is no longer possible. For $\gamma = .03$, this combined effect already explains as much as 84.2% of variance in final resonance values. The Resonance model clearly requires quite a delicate setting of its decay rate parameter, raising the question whether the optimal values found here are suitable for other cases of reinstatement as well.

Resonance values

After the first processing cycle, the total amount of resonance in the text network equals the number of items that received an initial signal, which is at most 18 for the captain text. During processing of a sentence, however, the resonance values increase without any theoretical upper limit. For instance, for $\gamma = .03$, the total amount of resonance reaches levels close to 10^{13}. This is caused by the fact that there are no negative values in W, which results in Equation 2.1 not allowing resonance to ever decrease. If the value of γ is small, as it needs to be for reinstatement to occur, decay is slow and the process may run for quite a large number of cycles. As a result, resonance values increase dramatically.

This causes a problem when resonance values are to be interpreted psychologically. Presumably, the resonance value of an item is meant to indicate its activation in the reader's mental representation of the text. However, this implies that the model predicts these activations to be approximately 10^{12} times larger after processing a sentence than at the start of processing. Since this is clearly not realistic, resonance values from different processing cycles cannot be compared to one another. As a result, the Resonance model cannot be used to track the mental activation of an item during processing, but only to compare its activation to those of the other items in the same processing cycle.

For larger decay rates, resonances obtain more reasonable levels. If $\gamma = .5$, for instance, the total amount of final resonance is at most 81.4. However, as discussed above, such large values of γ do not lead to reinstatement of any critical items.

2.1.4 Conclusion

The Resonance model suffers from a technical problem that has to be solved before it can be considered a robust computational model: The resonance values need to be limited to a fixed range. Furthermore, the decay rate γ needs to be shown appropriate not just for the captain text, but for any text that the model is to process. In Jacobs and Grainger's terms, the model's stimulus generality has to be established. Considering the narrow range of γ resulting in reinstatement of critical items in the original text but not in the alternative text, this may turn out to be problematic. Myers & O'Brien (1998, p. 148) did find the same decay rate of $\gamma = .0198$ appropriate for both the captain text and a second text, but a third text required a value of $\gamma = .0247$ (p. 151). The difference between these two decay rates may seem small, but it is in fact quite large when compared to the size of the range of γ we found to be appropriate for the captain text. This indicates that the decay rate may need to be adjusted for processing a new text. Technical issues aside, however, the model does show how bottom-up reinstatement of backgrounded material is possible in principle although the current example is by itself not very convincing.

Be reminded that the Resonance model was designed to explain the results of the experiment by Albrecht and Myers (1995) who found longer reading times on sentences that are inconsistent with earlier information when this earlier information was supposedly reinstated just before, compared to when it was not. They concluded that critical elements could be restored to working memory by a bottom-up process. Although the Resonance model is a simulation of this part of the comprehension process, the effect of reinstatement on reading times was not predicted by the model. In our simulations, a total of 1,977 processing cycles were needed to process the inconsistent sentences 16 and 17 of the original captain text. For the alternative text in which sentence 15 did not mention *large desk* and less reinstatement occurred, this number was the same.

Even if the model would show an effect of reinstatement on the number of processing cycles, this could not be a simulation of the same effect in human subjects. The slowdown of reading on an inconsistent sentence must be related to the reader's knowledge of the world, since only this knowledge defines whether or not the captain text is inconsistent in the first place. In order to detect the inconsistency, the reader must infer that the statement about the captain having to complete the inventory implies that FINISHED(CAPTAIN,INVENTORY) is not true. In conjunction with the meaning of sentence 3, about having to finish the inventory before shore leave can begin, it follows that CAN(BEGIN(CAPTAIN,LEAVE)) is false as well. Finally, the reader must know that this latter falsehood is inconsistent with the meaning of the last sentence, which states that the captain is ready for his shore leave. Each of these inference steps requires knowledge about the meaning of words and about relations among truth values of propositions. The model, however, uses no such knowledge. All network nodes follow directly from the text, and the links between them are based on formal, not semantic, considerations. Whether or not two items are connected depends only on their co-occurrence in a sentence and on propositional forms, but not on the items' relation to the reader's knowledge. Of Kintsch and Van Dijk's (1978) three levels of discourse representation discussed in Section 1.1.2, the Resonance model represents texts at the textbase level, while a situational representation is required to detect inconsistencies.

In theory, world knowledge can be included by letting propositions from the reader's knowledge resonate like text items. Myers and O'Brien claim that only practical considerations prevented them from implementing this:

> We believe that the propositions and concepts in the reader's general knowledge store also resonate and play an important role in processing. However, because of our inability to detail the contents of the knowledge store, we suffer the limitation of representing only the text. (Myers & O'Brien, 1998, Note 2)

However, even if a general knowledge network is available, adding it to the Resonance model may not be that simple. Readers have a large amount of world knowledge that will become massively connected to text items. The text concept SHIP, for instance, should be connected to everything that is known about ships. If signals are allowed to freely spread through the resulting huge network, any focus on the text will be lost. Somehow, only knowledge that is directly relevant to the text should be considered. This, of course, raises the problem of how to select text-relevant items from all the potentially relevant general knowledge. Some of these issues are treated by the Construction-Integration model discussed in Section 2.4.

2.2 The Landscape model

There is of course more to discourse comprehension than the fluctuating activations of text items as simulated by the Resonance model. Some of the higher-level aspects of discourse comprehension will be discussed in later sections. Here, we look at the construction of a relatively stable memory representation of the text, resulting from the comprehension process. The Landscape model (Van den Broek, Risden, Fletcher, & Thurlow, 1996) simulates how activations of text items lead to such a memory trace. It takes as input the activations of text items over a sequence of sentences and computes from this the strength of the items' retention in memory, and the strengths of the relations between them.

2.2.1 Model description

Activation values

During processing of sentence t of a text, any concept p is assumed to have an activation value $x_{p,t}$, indicating the extent to which the concept is available in the reader's working memory when that sentence is processed. In theory, propositions could also be included but we shall follow Van den Broek et al. (1996; Gaddy, Van den Broek, & Sung, 2001; Van den Broek, Young, Tzeng, & Linderholm, 1999) and restrict ourselves to concepts.

Note that the model does not explain the activation values, but that they form its input. For instance, they could follow from the Resonance model. Alternatively, a theory of inference can be made explicit by setting the activation values of non-text items accordingly. From these, the Landscape model constructs a memory representation that can be compared to empirical data in order to judge the validity of the inference theory.

According to Van den Broek et al. (1996, p. 171), there are three reasons why a concept can be activated during reading of a sentence. First, if concept p is stated in sentence t, it receives the maximum activation value of $x_{p,t} = 5$. For instance, from the sentence *A young knight rode through the forest*, Van den Broek et al. (1996, Tables 6.1 and 6.2) extract the concepts KNIGHT, RODE, and FOREST, which each get the maximum activation during processing of that sentence. Second, concepts can be inferred from the reader's world knowledge.

From the sentence about the knight riding through the forest, the two concepts HORSE and TREES are assumed to be inferred and get an activation value of 2. Inferences that are required for causal or anaphoric coherence receive an activation value of 3 or 4, thereby implementing the theory that these inferences are most important to discourse comprehension. Third, a concept that is mentioned or inferred in sentence t but not at $t + 1$, has a residual activation at $t + 1$ of $x_{p,t+1} = \frac{1}{2}x_{p,t}$. If p is not mentioned or inferred again at $t + 2$, then $x_{p,t+2} = 0$.

A three-dimensional surface plot of these input values vaguely resembles a mountain landscape. It is from this image that the model gets its name.

Strength values

Unlike most other models, the Landscape model does not include a process comparable to activation spreading. In fact, no iterative process for the integration of a sentence takes place at all. Instead, the text's memory representation after processing sentence t is computed directly from t's activation values and the previous memory representation.

During processing of the text, each concept p builds up a *strength* value s_p. Initially, all these values equal 0. Also, the *relation strength* w_{pq} between each pair of concepts p and q is 0 initially but builds up as text processing proceeds. After processing sentence number t, the strength of concept p and of the relation between p and q ($p \neq q$) are increased by

$$\begin{aligned}\Delta s_p &= x_{p,t}\\ \Delta w_{pq} &= x_{p,t}x_{q,t},\end{aligned}$$

that is, when a sentence is processed, concept strengths increase by the concept's activation in the sentence, and the strength of the relation between two concepts increases by the product of their activations (Van den Broek et al., 1996, p. 176).[3] Concepts that are often named in the text, or inferred from it, are active in many sentences and therefore end up with a large strength. Likewise, pairs of concepts that are often active together receive a large relation strength. In short, the Landscape model assumes that concepts receive a strong mem-

[3] Van den Broek et al. (1996) note that this algorithm is a simplification. In Van den Broek et al. (1999) and Gaddy et al. (2001), a more complex process called 'cohort competition' or 'cohort activation' is described informally, without enough details for a formal specification.

ory representation if they are often present in working memory, and that two concepts become strongly associated if they often co-occur in working memory.

2.2.2 Evaluation

With input activations set as described above, the Landscape model was able to predict results on a free recall task (Van den Broek et al., 1996, pp. 179-181; 1999, p. 85). After processing a text, the strengths of concepts, and of their relations to other concepts, predicted the probability that the concepts were recalled by subjects who read the same text. Moreover, the concept most likely to be recalled first was the one with the largest strength value. After recall of concept p, the concept that was most likely to be recalled next was the one for which the model predicted the largest relation strength with p.

Considering the model's simplicity, these results are not very surprising. Regular mention or inference of a concept can be expected to lead to a strong representation of the concept in memory, and therefore to a high probability that the concept is recalled. In the model, regular activation leads to large concept and relation strength values. Also, regular co-occurrence of two concepts is likely to result in a strong association between them in the text's memory representation, and therefore to a high probability that the one is recalled directly following the other. In the model, co-occurrence of concept activations leads to a strong relation between the two concepts. In short, the similarities between the model's results and empirical data say more about the appropriateness of the assumed activation values than about the quality of the Landscape model.

2.2.3 Conclusion

Given appropriate activation values of text items, the Landscape model computes the strengths of the items, and of the relations between them, in the memory representation of the text. The correspondence between the model's result and empirical data shows that the model captures at least some part of the memory representation that readers actually create. However, the model can be said to suffer from a lack of explanatory adequacy. It merely formalizes the well-known fact that simultaneous activation of concepts in working memory can lead to a long-term association of these concepts. The model does not explain where these activations come from, how they can lead to activation of concepts from the reader's world knowledge, or why there is a relation between associative strengths and order of recall.

Furthermore, a story is more than a collection of concepts and associative relations. Readers are also able to recall the sequence of events described by the text, but the Landscape model does not allow for such a representation if it incorporates only concepts. Of course, activation values of propositions corresponding to the story's events can be added to the concept activations, but for these propositions, too, the relation strengths only reflect co-occurrence in working memory. For adequate story comprehension, causal relations between story events are more important to encode than co-occurrence relations. Finding such causal relations in a story, however, requires the model to have knowledge about causality in the world and the Landscape model has no such knowledge. Alternatively, causal connections instead of activation values could serve as model input. In the next section, a model is discussed that uses this type of input to compute hypothetical mental representations of stories.

2.3 The Langston and Trabasso model

One of the most important factors influencing story comprehension is the causal relatedness between the story's statements. Statements that have a stronger causal relation to previous story events are read faster (Myers, Shinjo, & Duffy, 1987), recalled more often (Myers et al., 1987; Trabasso & Van den Broek, 1985), and rated more important to the text (Trabasso & Sperry, 1985). Moreover, when a statement is read, the story events to which it is causally related become more available to the reader (Suh & Trabasso, 1993; Lutz & Radvansky, 1997).

Neither the Resonance model nor the Landscape model incorporates causality, so they cannot account for these results. Langston and Trabasso (1999; Langston, Trabasso, & Magliano, 1999) developed a causality-based model that is to simulate all of these effects.

2.3.1 Model description

The text network
There are two important similarities between the Langston and Trabasso model and the Resonance model. First, sentences are processed one at a time. Second, a network of text elements is constructed. In the Langston and Trabasso model, however, this network does not contain any concepts or propositions. Instead, each network node corresponds to one sentence from the story. Another difference is that the network connections are based on causal relations between the events described by the sentences. Two sentence nodes p and q are causally connected if the sentences' events pass the so-called *counterfactual test* (Langston & Trabasso, 1999, p. 35): If q would not have occurred without p (all other things being equal), and there is no intervening event caused by p and causing q, then p and q are causally connected.[4]

Table 2.3 shows the 'Ivan text' used by Langston and Trabasso (1999) to test their model. The corresponding text network is shown in Figure 2.3. Although the rules for deciding upon causal connections seem quite clear, the relation

[4] Langston et al. (1999) also used a version of the model in which nodes were connected by argument overlap, as in the Resonance model. Since they found that this alternative model predicted less empirical data than the causally connected model (p. 222), we shall not discuss the argument-overlap model.

Table 2.3: The sentences of the Ivan text, corresponding to the network in Figure 2.3 (adapted from Langston & Trabasso, 1999, p. 36, and Langston et al., 1999, Table 6.1).

t	sentence
1	Ivan was a great warrior.
2	One day, Ivan heard that a giant had been terrifying people in his village.
3	Ivan was determined to kill the giant.
4	When the giant came, Ivan shot an arrow at him.
5	Ivan hit him but the arrow could not hurt the giant.
6	One day, a famous swordsman came to a nearby village.
7	Ivan decided to learn how to fight with a sword.
8	He went to the swordsman.
9	Ivan studied hard for several weeks.
10	He became a very skilled swordsman.
11	That night, Ivan returned home to his village to find the giant.
12	Ivan attacked the giant.
13	Ivan finally killed the giant with his sword.
14	The people thanked Ivan a hundred times.

between the sentences and the network is not always obvious. For instance, sentences 1 and 2 should probably not pass the counterfactual test: If Ivan had not been a great warrior, he would nevertheless have been likely to hear about the giant. Furthermore, the network shows a causal connection between sentences 3 and 5. Indeed, if Ivan had not wanted to kill the giant, the giant would not have been hit by an arrow. However, sentence 4 seems like a clear intervening event: Ivan shoots the arrow because he wants to kill the giant, which causes the giant to be hit. In spite of such problems, the text network of Figure 2.3 was used in our simulations.

When a sentence is read, its node is added to the text network. The connection weights between this node and the others in the text network depend on their causal connections. To be exact, the weight of the connection w_{pq} between p and q equals 7 minus the number of causal connections in the shortest path between p and q in the text network, with a minimum of 0 (Langston & Trabasso, 1999, p. 36). In practice, this means that

- All nodes are connected to themselves with the maximum weight of 7 ($w_{pp} = 7$).
- The connection weight matrix is symmetrical, so $w_{pq} = w_{qp}$.
- If p and q are causally connected, the weight of the connection between them is $w_{pq} = 6$. These are the only connections shown in Figure 2.3.

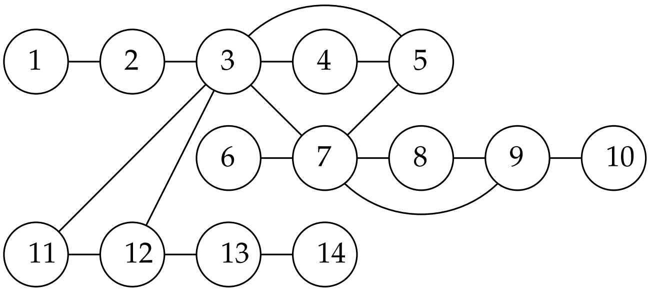

Figure 2.3: Complete text network of the Ivan text (Langston & Trabasso, 1999, Figure 2.2). Node numbers refer to the sentences in Table 2.3 and indicate the order in which the nodes and their connections enter the model. The links between nodes indicate direct causal connections.

- If there is a path between p and q, but they are not causally connected directly, $0 \leq w_{pq} \leq 5$.
- If there is no path between p and q, $w_{pq} = 0$.

For example, of all shortest paths between nodes in the Ivan network, the longest is the one between nodes 10 and 14. It takes at least 6 steps to get from one to the other ($10 \rightarrow 9 \rightarrow 7 \rightarrow 3 \rightarrow 12 \rightarrow 13 \rightarrow 14$), so their initial connection weight is $w_{10,14} = w_{14,10} = 7 - 6 = 1$.

The integration process

After determining the weights of the connections of the new sentence node, an integration process takes place that updates the connection weights (Langston & Trabasso, 1999, pp. 39-40). This starts with assigning to each node p a positive activation value x_p. The new sentence node has an initial value of $x_p(0) = .5$, and all other nodes begin with half the value that resulted from the integration process of the previous sentence. Next, a two-step activation spreading process is applied repeatedly. In the first step, each node p receives an intermediate activation value x'_p that equals the sum of the values of all nodes, weighted by their connection to p:

$$x'_p(c) = \sum_q x_q(c) w_{pq}.$$

Next, the activation values are normalized by dividing them through the sum of all intermediate activation values, resulting in a total activation of 1:

$$x_p(c+1) = \frac{x'_p(c)}{\sum_q x'_q(c)}. \tag{2.3}$$

This process is repeated until the activation values no longer change very much. According to Langston and Trabasso, this is the case when

$$\sum_p x_p(c+1) - \sum_p x_p(c) < \theta, \tag{2.4}$$

with θ an arbitrarily small but positive value. It is clear that this cannot be the stopping criterium that was actually applied. Because of normalization (Equation 2.3), the sum of all activation values equals 1 after every processing cycle. Therefore, the change in total activation expressed in Equation 2.4 is always 0 and the process will halt immediately. It is likely that not the *change in total activation* was taken as a criterium, but the *total change in activation*.[5] This is expressed by the equation used in our simulations:

$$\sum_p \left| x_p(c+1) - x_p(c) \right| < \theta. \tag{2.5}$$

The value of the parameter was set to $\theta = .001$.

Updating the connection weights

After activation has settled (i.e., Equation 2.5 is satisfied), the connection weights are updated. Each weight is increased by an amount equal to the product of its current weight and the activation values on both ends of the connection:

$$\Delta w_{pq} = w_{pq} x_p x_q. \tag{2.6}$$

When node number 1 enters the model, it necessarily receives all activation since there are no other nodes, resulting in $x_1 = 1$. Since its only connection is the one to itself, with an initial weight of 7, the weight increase equals $\Delta w_{1,1} = 7$ and the updated weight becomes $w_{1,1} = 14$. Assuming that the second node is causally connected to the first, the initial weight of this connection is $w_{1,2} = w_{2,1} = 6$. Of course, the second node is also connected to itself with $w_{2,2} = 7$. The first node's self-connection weight is larger than the sec-

[5] Actually, Langston and Trabasso (1999, p. 40; Langston et al., 1999, p. 190) claim that Equation 2.4 expresses the total change in activation values.

ond node's, so the first will receive more activation. In fact, after activation has settled, the vector of activation values equals $X = (.64, .36)$. As a result, the first node's self-connection weight increases more than the second node's. This effect is amplified because in Equation 2.6 the increase in connection weight is multiplied by the weight itself.

It is not hard to see that no connection weight can catch up with the head start of the first node's self-connection. After processing the Ivan network we found that the largest weight was the first node's self-connection weight: $w_{1,1} = 66.1$. The second-largest weight was the one between the first two nodes and had a much smaller value of $w_{1,2} = 13.6$. Not only are such results unrealistic, they also differ from the numbers given by Langston and Trabasso (1999, Figure 2.3). We found that those data could not be replicated unless the 'head start'-effect was cut down by making all self-connection weights non-adjustable. In other words, although this is not mentioned anywhere, it seems like Equation 2.6 is only valid for $p \neq q$.

2.3.2 Evaluation

The model's results are claimed to account for a large variety of empirical data: reading times, judgments of importance and relatedness, naming and verification times, and recall probabilities. In all these cases, the data were predicted by the connection weights W. This is not very surprising, since such data are known to depend strongly on causal relatedness, which is encoded in the network's initial connection weights. Therefore, in order to test the model's ability to predict empirical data, it is irrelevant that the connection weights after (or during) story processing account for the data. Instead, it needs to be shown that they predict the data better than the initial connection weights do. However, nowhere do Langston and Trabasso show that this is indeed the case.

We found that 86.2% of variance in final connection weights of the Ivan network was accounted for directly by the initial weights that constitute the model's input (self-connections were ignored, because they are never adjusted). This shows that the model does not change the connection weights very much. The empirical data can therefore not be expected to be predicted much better by the model's output than by its input.

What causes the small difference between the initial and final weights? Since connection weights and activation values are always positive, it is immediately clear from Equation 2.6 that weights can never decrease. This results in a primacy effect: The longer a connection is in the model the larger its weight will become, so earlier sentences receive larger connection weights. This effect reinforces itself, because the rise in connection weights (Equation 2.6) increases with larger weights. Moreover, the nodes that are connected with larger weights receive more activation, which increases Δw_{pq} even more. Primacy[6] accounted for 44.5% of variance in final connection weights. Taken together, 95.3% of variance in final connection weights was explained by initial weights and primacy. In other words, the computational model does not do much more than take the input connection weight matrix and increase the weights of earlier nodes.

2.3.3 Conclusion

The Langston and Trabasso model takes causal connections between story sentences and increases the importance of the connections between earlier sentences. Apart from the reasonable question whether such a simple operation is worth implementing as an iterative process and to be called a model, the resulting primacy effect is not even helpful. Langston and Trabasso (1999) note that "the general tendency for later sentences to be lower in connection strength leads to underestimation of empirical data" (p. 63). Since all the model accomplishes is this unwanted primacy effect, it can be expected that empirical data would be predicted more accurately *without running the model*.

This raises the question which mental process the computational model is meant to simulate. If the number of cycles to settle had predicted reading times, the model's process might be claimed to simulate part of the reading process. However, model processing time was not found to be related to any empirical observation. All in all, the Langston and Trabasso model does not add anything to a simple analysis of causal relations in a story. So what is the model's purpose? The answer is given by Langston and Trabasso (1999):

> We advocate and use a discourse analysis . . . to identify a priori causal connections that *could be made by the readers* during the processing of a discourse. We use a connectionist model to simulate how people might

[6] The primacy of a connection is defined as the logarithm of the number t of the sentence that brought the connection into the network. Using the logarithm reduces the importance of more recent connections, thereby compensating for the self-reinforcing effect of primacy.

> use their "expert" knowledge of psychological and physical causation to make these causal connections during understanding. (p. 33)

The model is claimed to simulate the making of causal connections. However, before it is run, the modeler needs to identify the "causal connections that could be made by the readers" to serve as input to the computational model. Next, the integration process is supposed to select which of these possible connections are actually made. This is a rather curious division of labor, since identifying all possible causal connections between story sentences requires a substantial amount of reasoning with world knowledge. It seems unlikely that readers will first do all this work to find causality in a text, just to be able to ignore most of it. Therefore, the discourse analysis is probably not meant to be part of the model but is simply some pre-processing necessary to create useful input. However, this means that the model can only process texts in which all possible causal relations are stated explicitly. In practice, such texts are quite rare.

2.4 The Construction-Integration model

All of the models discussed so far ignore one aspect that is vital for a full account of discourse comprehension: the selection of world knowledge relevant to the text. In the Resonance and Landscape models, propositions and concepts that do not originate from the text have to be supplied by the modeler. The same is true of the knowledge about causal relations that forms the input to the Langston and Trabasso model.

Combining a computational model with world knowledge is problematic for at least two reasons. First, the amount of world knowledge readers have is simply too large to implement any significant part of. Second, even if a fairly large amount of world knowledge were to be implemented, a model of discourse comprehension should explain how the text-relevant part of this knowledge is selected.

The Construction-Integration model (Kintsch, 1988, 1998) makes a beginning at handling these problems. It assumes that the reader's world knowledge is stored in a so-called *knowledge net*, consisting of concepts and propositions connected to one another by weighted links. No attempt is made to actually implement a substantial part of this net. Instead, when a text is processed, the concepts and propositions from the text select some associated concepts and propositions from the hypothetical knowledge net, and the rest of the net is ignored. Next, from the resulting collection of items, only the most relevant ones are kept while the others are discarded. These two processes take place in two separate phases. In the first phase, called *construction*, items from the knowledge net are selected. Next, less relevant or inappropriate items are discarded in the *integration* phase.

2.4.1 Construction

The construction phase takes as input a collection of concepts and propositions that correspond to the sentence being processed. For example, consider the text *The lawyer discussed the case with the judge. He said "I shall send the defendant to prison."* Note that the pronoun *he* is ambiguous: It can refer to either the lawyer or the judge. Of course, knowledge about the legal system tells us that *he* must be the judge, since lawyers do not send people to prison.

According to Kintsch (1988), this text can be parsed into the five text propositions in Table 2.4. Two of these correspond to the incorrect interpretation in which *he* refers to the lawyer, and two others correspond to the correct reading in which *he* is the judge.

Table 2.4: Five propositions from the text *The lawyer discussed the case with the judge. He said "I shall send the defendant to prison."* (Kintsch, 1988, p. 169).

label	text proposition
T1	DISCUSS(LAWYER,JUDGE,CASE)
T2	SAY(LAWYER,T3)
T3	SEND(LAWYER,DEFENDANT,PRISON)
T4	SAY(JUDGE,T5)
T5	SEND(JUDGE,DEFENDANT,PRISON)

Construction now consists of three steps: association, inference, and connecting. The result of all this is a network of concepts and propositions, connected to each other by weighted links. This network is called the *enriched textbase* (Kintsch, 1988, p. 166).

Association

Each concept and proposition from the text retrieves a small number of items from the knowledge net. In this knowledge net, items p and q are connected by a link with a weight s_{pq}, ranging from -1 to $+1$.[7] If item p also occurs in the text, the probability that it retrieves knowledge net item q depends on the weight s_{pq} between the two items in the knowledge net.[8] If $s_{pq} \leq 0$, item q is not retrieved by p. Otherwise, the probability that q is retrieved by p equals

$$\Pr(q|p) = \frac{s_{pq}}{\sum_r s_{pr}} \tag{2.7}$$

where r ranges over all nodes in the knowledge net positively connected to p (Kintsch, 1988, p. 166). Of course, Equation 2.7 is not applied in practice because the 'real' knowledge net connection weights are unknown and estimating the weights between p and all nodes in the knowledge net is impossible. Still, the

[7] That is, according to Kintsch (1988). Other values are often used, see the evaluation in Section 2.4.3.

[8] This presupposes that any possible (or at least any sensible) text proposition is already present in the knowledge net, which might seem hard to believe but since, in practice, connection weights are simply chosen by the modeler, it does not need to worry us much.

basic idea behind the association step is clear: Each text item retrieves a few items from the knowledge net, and knowledge net items have a better chance at being retrieved if their connection to the text item is stronger.

It is important to note that nothing but connection weight has an effect on the stochastic retrieval process. It is not possible for context information to influence which knowledge net items are chosen. As Kintsch (1988) puts it: "The construction process lacks guidance and intelligence; it simply produces potential inferences, in the hope that some of them might turn out to be useful" (p. 167).

The number of knowledge net items retrieved by each text item is claimed to be approximately 5 to 7, but when the Construction-Integration model is actually brought into practice, this number usually ranges from 0 to about 3. The five text propositions in Table 2.4, for instance, are assumed to retrieve only two knowledge net propositions in total. Text proposition SEND(LAWYER,DEFENDANT,PRISON) (T3) does not retrieve any items, since the reader does not know anything about lawyers sending defendants to prison. On the other hand, the proposition SEND(JUDGE,DEFENDANT,PRISON) (T5) does retrieve two items from the knowledge net. These propositions, shown in Table 2.5, are (A1) SENTENCE(JUDGE,DEFENDANT) and a proposition (A2) linking A1 and T5 by stating that sentencing the defendant implies sending him or her to prison.

Table 2.5: Two propositions from the knowledge net, retrieved by the propositions in Table 2.4. (Kintsch, 1988, p. 169).

label	associated proposition
A1	SENTENCE(JUDGE,DEFENDANT)
A2	IMPLY(A1,T5)

Inference

In case a proposition is required for comprehension of the text, but is neither in the text nor did it result from the association step, an inference step allows for a focused search for propositions in the knowledge net (Kintsch, 1988, p. 167). This part of the construction phase is sometimes claimed to be skipped, usually just ignored, and never actually implemented. However, as shall be argued later (see Section 2.4.4) it is often necessary for finding required inferences.

Connecting

Up to now, the construction process resulted in a collection of unconnected concepts and propositions, originating both from the text and from the knowledge net. Next, these items are connected to each other by assigning a weight w_{pq} to each pair of items p and q from the enriched textbase. Two items that both originate from the text are connected with a positive weight "proportional to their proximity in the text base" (Kintsch, 1988, p. 167). How this proximity is determined remains unclear. If one item (or both) originated from the knowledge net, the connection weight is inherited from there: $w_{pq} = s_{pq}$ (Kintsch, 1988, p. 167). Of course, the 'real' weights from the knowledge net are unknown, so in practice it is up to the modeler to choose values as he or she sees fit. The result of all this is a weight w_{pq} for each pair of items p and q in the enriched textbase. Taken together, they form the connectivity matrix W.

The enriched-textbase network resulting from the 'lawyer and judge' example, including connection weights, is shown in Figure 2.4. The positive connection weights between text propositions depend on their distance in the text. The connections to, and between, associated propositions are assumed to have a weight of .5. Propositions corresponding to different interpretations of the pronoun are maximally negatively connected, since they exclude each other.

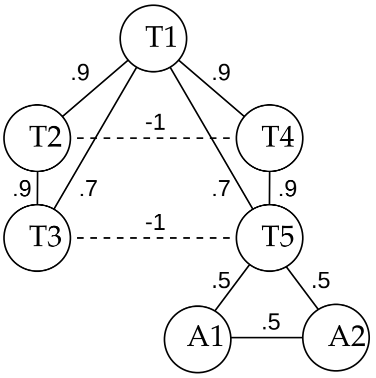

Figure 2.4: Enriched textbase network corresponding to the propositions in Tables 2.4 and 2.5. The values are the connection weights, which are symmetrical so $w_{pq} = w_{qp}$. Dashed lines indicate negative connections (Kintsch, 1988, pp. 169-170).

2.4.2 Integration

The integration phase of the Construction-Integration model takes as input the enriched textbase that resulted from the construction phase and selects which nodes of this network can be discarded because they are less relevant to the text than others, or because they are inconsistent with the rest of the enriched textbase. This is accomplished by assigning to every item p an activation value x_p. Initially, all items that originated from the knowledge net have an activation value of $x_p(0) = 0$. Among the items originating from the text an initial total activation of 1 is divided equally. Next, the integration process iteratively applies a three-step algorithm to the activation values: spreading activation, discarding negative values, and normalization (Kintsch, 1988, p. 168). The only difference between this integration process and Langston and Trabasso's (see Section 2.3.1) is that negative values are discarded in the Construction-Integration model. However, since all initial activations and all values in matrix W of the Langston and Trabasso model are positive, negative values can never arise in that model.

Spreading activation The vector $X(c)$, containing the activation values at processing cycle c, is multiplied by the connectivity matrix W, thereby spreading the activation of each node to its neighbors. This means that every node's activation value is replaced by an intermediate value that equals the sum of the activation values of all nodes, multiplied by the connection weights:

$$x'_p(c) = \sum_q x_q(c) w_{pq}.$$

Discarding negative values Negative activation values are set equal to 0, resulting in the new intermediate activation values

$$x''_p(c) = \max\{x'_p(c), 0\}.$$

It is not clear why activation values are not supposed to be negative, although this is necessary to make the following step function properly.

Normalization To prevent activation values from increasing towards infinity, the intermediate activation values are normalized to sum up to 1, resulting in the new activation values[9]

$$x_p(c+1) = \frac{x''_p(c)}{\sum_q x''_q(c)}.$$

If the average difference between the activation values $x_p(c)$ and $x_p(c+1)$ is larger than some small fixed value θ (usually, $\theta = .001$) the three steps are repeated starting with $X(c+1)$. Otherwise, the integration process is completed. The most relevant items are now assumed to have the highest activation value, because they have a more central position in the network.

In the 'lawyer and judge' example, proposition T5 has more positive connections than its competitor T3 and will therefore receive a higher activation value during integration. As a result, proposition T4 is connected to a more active node than its competitor T2, so it too will receive more activation. Eventually, in our simulations of the integration process, the activations of T2 and T3 (the LAWYER-nodes) become 0, while the activations of T4 and T5 (the JUDGE-nodes) remain high, as shown in Figure 2.5. The incorrect interpretation of the pronoun is discarded, while the correct one is kept: It is the judge, and not the lawyer, who says that he shall send the defendant to prison.

2.4.3 Evaluation

The construction of an enriched textbase network is largely a subjective task. It is up to the modeler to decide which knowledge items are associated with the text items, and this decision is rarely a fair one. For example, Schmalhofer, McDaniel, and Keefe (2002) used the Construction-Integration model to explain how inferences are made to increase coherence between sentences. They let the model process two different texts, both starting with the sentence (1) *The director and the cameraman were preparing to shoot closeups of the actress on the edge of the roof of the 14 story building when suddenly the actress fell*. The next sentence was either (2a) *Her orphaned daughters sued the director and the studio for negligence* or (2b) *The director was talking to the cameraman and did not see what happened*. From (2a) it can be inferred that the actress died, but from (2b) it cannot. Indeed, the simula-

[9] In Kintsch (1998), these values are normalized so that the maximum value equals 1. The difference between the two kinds of normalization lies only in the scaling of activations. The values do not change relative to one another.

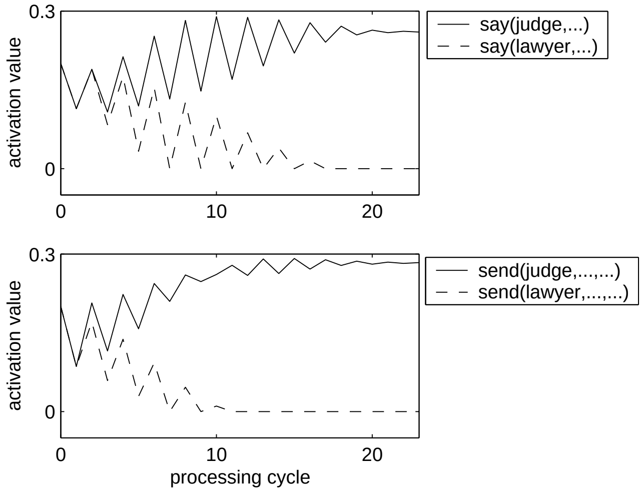

Figure 2.5: Activation values of SAY (above) and SEND (below) propositions, during integration of the network of Figure 2.4.

tions resulted in a high activation value for the proposition DEAD(ACTRESS) after processing of (2a), but not after (2b). However, to make this high activation value possible, the proposition DEAD(ACTRESS) has to be part of the enriched textbase even though it is not part of the text. Therefore, this proposition was added during the construction phase of sentence (1). No other proposition was added. Of course, this was because the modeler knew that one of the sentences (2a) and (2b) implies that the actress died. But what if the next sentence had turned out to be *She was released from hospital after two weeks*? In this case, the proposition WOUNDED(ACTRESS) should have been added to the textbase in the construction phase of sentence (1). And how about *The stunt coordinator was very pleased with her practice jump*? In short, any possible outcome of sentence (1) needs to be selected during the construction phase to make its inference possible. It is up to the modeler to choose at least the propositions that are expected to be inferred later.

Considering the huge amount of general world knowledge people have, the only way to incorporate some of it in a computational model seems to be to subjectively choose a subset that is small enough to handle but large enough to require a construction phase. It should then be left to the model to decide which part of the implemented knowledge is relevant to comprehension of the text. This is how the Construction-Integration model should be used in practice. However, it hardly ever is. Usually, the modeler chooses exactly the items necessary for comprehension, and just a few distracting items (or, like Schmalhofer et al., none at all). In effect, the modeler is executing the inference step of the construction phase, and not the 'guidance and intelligence'-lacking association step.

Setting the weights of connections between textbase items is another source of subjectivity. There are no general guidelines (let alone rules) for deciding what the weight of each connection should be. The only constraint according to Kintsch (1988) is that weights range from -1 to $+1$, but Kintsch, Welsch, Schmalhofer, and Zimny (1990) used integer values from 0 to 5. Singer (1996) applied the model using only weights of 0 and 1, while Tapiero and Denhière (1995) included weights of -3 and 2. Also, it is not clear whether items should be connected to themselves. In Kintsch (1988) they are not, but according to Kintsch (1998) they are. Kintsch (1992) even uses the weights of these self-connections to indicate which propositions are emphasized by the text. Moreover, connectivity matrices always turn out to be symmetrical, but nowhere is it stated that this should be the case.

In short, there is too much freedom in setting connection weights. This is especially harmful since the properties of the integration process depend heavily on these weights. Not only do they influence the final result, but integration itself may not make sense if connection weights are chosen wrongly. For instance, it is not known under which circumstances the integration process is well defined (i.e., does not result in all activations being zero). Rodenhausen (1992) showed that the integration process is well defined if the connectivity matrix is symmetrical and, for every node, the sum of the connection weights from the node to all others is strictly positive, but it remains unclear what happens when a node has a negative weight sum.

Another issue concerning the integration process is that it is not guaranteed to converge to a stable interpretation. It is still an open question under which conditions the process converges. Guha and Rossi (2001) claimed that, for any number of items in the enriched textbase, it is possible to construct a connectivity matrix that does not result in a converging process.[10] Such a con-

[10] Guha and Rossi proved that if a matrix leads to a well defined integration process, this process converges if and only if the matrix' eigenvalue with the largest absolute value is positive. If

nectivity matrix does not even need to be unrealistic. Just a small change in the connection weights from Figure 2.4 will do the trick. When $w_{\mathrm{T1,T3}}$ and $w_{\mathrm{T1,T5}}$ are lowered from .7 to .45, and $w_{\mathrm{T2,T3}}$ and $w_{\mathrm{T4,T5}}$ are lowered from .9 to .6, the integration process never reaches a conclusion. After 30 cycles, the activation vector is still oscillating between two states (see Figure 2.6) and it will keep doing this for ever. This is in sharp contrast with the claim that in this very example

> other assignments [of connection weights] result in different numerical values for the final activation vector, but its pattern remains the same as long as the essential features of the matrix are preserved—for example, which connections are positive, negative, and zero. (Kintsch, 1988, p. 169)

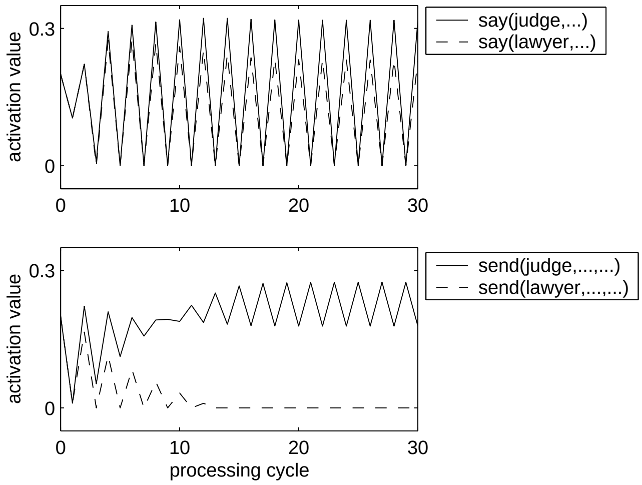

Figure 2.6: Activation values of SAY (above) and SEND (below) propositions, during integration of the network of Figure 2.4 with connection weights changed to $w_{\mathrm{T1,T3}} = w_{\mathrm{T1,T5}} = .45$ and $w_{\mathrm{T2,T3}} = w_{\mathrm{T4,T5}} = .6$.

It is very well conceivable that a reader's interpretation of a text oscillates between different possibilities, so convergence of the integration process is not

more eigenvalues share the same largest absolute value, they should all be positive. However, there are no guidelines for the construction of a useful connectivity matrix that satisfies this constraint.

strictly necessary. However, take another look at Figure 2.6. There is an oscillation between two interpretations: Although it is inferred that it is the judge who will send the defendant to prison, it remains unclear whether the judge or the lawyer *says* he will do this. On closer inspection, it can be seen that the activations of SAY(JUDGE,...) and SAY(LAWYER,...) oscillate *in phase*. Therefore, both interpretations are senseless: Either *both* the lawyer and the judge say that they will send the defendant to prison, or *neither* says so. The same is true in the early stages of processing the original connectivity matrix (Figure 2.5). Only after more than five processing cycles, the oscillation between two senseless interpretations decreases and one of the interpretations is clearly chosen over the other.

It is not hard to see the cause of this strange behavior. In every processing cycle, the activation value of every node is *replaced* by the weighted sum of its neighbor's values. Both the LAWYER and the JUDGE nodes have a high initial activation value. Since the two pairs are negatively connected, they switch each other off after the first processing cycle, resulting in low activations for both pairs of nodes. Next, these almost inactive nodes send one another just a small amount of activation but all four receive activation from the DISCUSS node (T1), resulting in more activation for both pairs of nodes in the next cycle.

If nodes are connected to themselves, each node sends itself its own activation, which results in the original activation not being replaced but *adjusted*. Figure 2.7 shows how these extra connections smooth the integration process in the current example. This again demonstrates that the connection weights not only determine the outcome of the integration process (as they should), but also have a strong influence on the properties of the integration process. As long as it is not known under which conditions the connection weights will smoothly lead to a (usually) stable interpretation, finding a working connectivity matrix requires a lot of tinkering.

2.4.4 Conclusion

Both the construction and the integration phase have been shown problematic. For integration, these problems are mostly technical and are likely to be overcome by changing the algorithm into a more reliable one. There seems to be no reason why the integration algorithm has to be exactly the one described above. The construction phase, on the other hand, suffers from a problem that

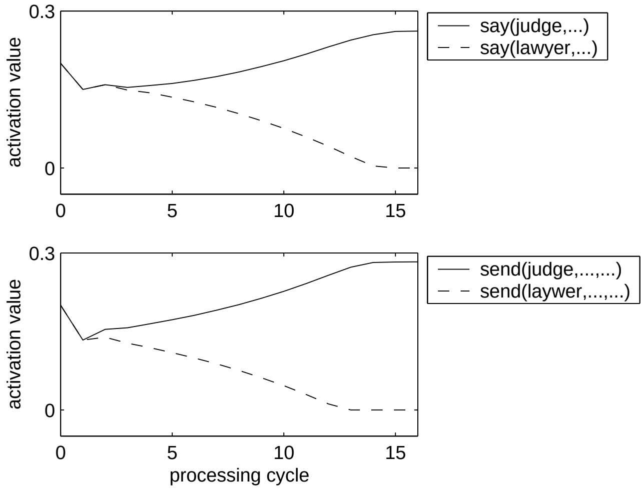

Figure 2.7: Activation values of SAY (above) and SEND (below) propositions, during integration of the network of Figure 2.4, when nodes are connected to themselves with a weight of 1.

is harder to solve: its subjectivity. In theory, only the connection weights between text items and knowledge net items have an effect on the choice of associated knowledge net items, but in practice, the modeler intervenes and selects items that are expected to be inferred later. In theory, connection weights follow from the knowledge net, but in practice, the modeler chooses them as he or she pleases. For a computational model, this is of course unacceptable. Kintsch (1998) discusses how some of this subjectivity can be overcome by extracting connection weights objectively from large text corpora, but, as explained in the following section, this method cannot be applied to obtain weights of connection between propositions. It can therefore not be said that the Construction-Integration model gives an explanation of how a text selects relevant items from world knowledge.

This critique does not concern the basic idea behind the model, namely that comprehension takes place in two distinct phases. In the first phase, text items retrieve associated items from world knowledge, without caring about context, being unambiguous, or making sense. In the second phase, the collection of items is pruned, resulting in a connected set of the most relevant items. All that was shown here is that the actual implementation of this model is never computational and objective. The question therefore remains: In the hypothetical case that a real knowledge net is available, can discourse comprehension be explained by a context-independent construction phase in which a few items are associated with the text, and a following integration process in which irrelevant items are discarded?

Consider the following sentence, adapted from Till, Mross, and Kintsch (1988): *The townspeople were amazed to find that everything had collapsed except the mint.* It seems obvious that *mint* refers to the building and not the candy.[11] Kintsch (1988) shows that the Construction-Integration model can come up with this correct interpretation by associating with the text concept MINT two knowledge net concepts: CANDY and BUILDING. Since the rest of the sentence is more related to buildings (which, unlike candies, can collapse), the concept BUILDING will receive more activation during the integration process than will the concept CANDY. In this way, the intended meaning of *mint* is obtained, while the incorrect meaning is discarded.

However, imagine that the sentence was part of a story about a town with an economy based on the trade in agricultural products. One day, the town's many traders expect a crash in the price of all herbs like rosemary and oregano. As the markets open, the townspeople are amazed to find that everything had collapsed except the mint.

In this context, mint is neither a candy nor a building. Since the 'herb'-context is clearly possible and comprehensible, the HERB concept should be associated with the text item MINT during the construction process, together with the CANDY and MONEY concepts. In other contexts, other concepts or propositions related to MINT may be required for comprehension, so these should also be added in the construction phase. One may wonder if there are any possible associates (i.e., positively connected knowledge net items) of MINT that need *never* be associated with it, that is, for which there exists no context in which such a possible associate is relevant to MINT. Well, in that case, *it is not a possible associate* and should not be positively connected to MINT in the knowledge net. In short, any text item needs to retrieve all knowledge items to which it is positively connected, because these are exactly the items for which there exists a context that makes them relevant. The resulting enriched textbase will necessarily become huge, needing a very powerful integration mechanism to

[11] The sentence actually used by Till et al. and by Kintsch (1988) did not read *everything collapsed* but *all the buildings collapsed*, making the meaning of *mint* even clearer.

select the relevant items. If such a hypothetical integration mechanism exists, why would it not be able to handle the entire knowledge net itself, making the construction phase redundant?

In short, a context-independent construction phase requires an integration process that is so powerful that the need for the construction phase itself is questionable. Alternatively, the construction phase may not be fully context-independent, resulting in a textbase that is only slightly enriched and can be processed by a fairly simple integration algorithm. However, if such an intelligent construction phase exists, why does it come up with some irrelevant items for the integration process to discard?

It seems unlikely that construction and integration can be strictly separated, and accepting a less strict separation begs the question: Why assume two phases anyway? A more realistic approach might be to assume a single process somewhat like spreading activation that is powerful enough to handle the complete knowledge net. Text items activate corresponding nodes in the knowledge net, and activation spreads from there. At first, this spreading activation process is context independent, resulting in activation of all items positively connected to the text items. However, as the process continues, activation more and more regroups to a path connecting text, context, and knowledge items, thereby linking the text to the previous context and associated world knowledge. Such a model is reminiscent of marker passing models (e.g., Charniak, 1983; Norvig, 1989), in which paths among nodes in a network are found by a single process that combines 'dumb' activation spreading (comparable to construction) and 'clever' search (comparable to integration).

Interestingly, Kintsch seems to have acknowledged that a context-independent construction phase will often not come up with a necessary proposition, and added the inference step that makes more focused problem solving possible within the construction phase. This is exactly the process that the Construction-Integration model is claimed to explain in the first place. However, when an actual text comprehension problem is implemented in the Construction-Integration framework, it is the modeler, and not the model, that performs the inference process and comes up with exactly the right proposition. How this inference step can be modeled computationally is as yet a mystery.

2.5 The Predication model

In the previous section, it was discussed that one of the problems facing the Construction-Integration model is the subjectivity in assigning weights to connections of the textbase network. Some of these weights supposedly reflect associative strengths in the reader's world knowledge network. Since these strengths cannot be obtained objectively, they need to be estimated by the modeler when implementing the Construction-Integration model. The Predication model (Kintsch, 2001) offers a partial solution to this problem, involving a switch from a localist to a distributed representation of words and propositions.

Up to now, only localist models were discussed. As was explained in Section 1.2.1, these models represent a piece of discourse as a network consisting of connected nodes. The nodes represent discourse items (like concepts or propositions) and the connections stand for relations (causal, associative, or otherwise) between them. In a localist model, therefore, items and relations are represented separately. In a distributed representation, on the other hand, there is no clear distinction between items and relations. Items are represented as vectors, and these vectors themselves reflect the relations among the items. For the Construction-Integration model, this changes the issue of choosing connection weights into one of choosing vectors representing concepts and propositions.

For words, there exist several methods to acquire vector representations automatically and objectively. The most influential of these methods is known as Latent Semantic Analysis (Landauer & Dumais, 1997). The Predication model is meant to describe how vectors representing propositions[12] can be constructed from the word vectors that result from Latent Semantic Analysis.

2.5.1 Latent Semantic Analysis

Altmann (1997) claims that "at its simplest, the meaning of a word is the knowledge that one has of the situations or contexts in which it would be appropriate to use that word" (p. 120). Latent Semantic Analysis (LSA) is nothing more than a formalization of this idea. It takes as input a large number of text sam-

[12] It is not always clear whether the Predication model finds vector representations for propositions or only for the proposition's predicate. For now, it is assumed that Predication's vectors do represent propositions. We shall get back to this issue in the conclusion (Section 2.5.5).

ples and lists how often any word occurs in these samples. If a word occurs in a certain sample, that must mean that it is "appropriate to use that word" in the "situation or context" formed by the text in that sample.

All results in this section, as those of Kintsch (2001), are based on 37,651 text samples of general reading for an American school child from grade 3 up to its first year in college (see the LSA website at lsa.colorado.edu). This corpus consisted of 92,409 word types (approximately 11 million tokens). The first step in performing LSA is the creation of a matrix with a row for each word type and a column for each sample. The values in the matrix are the number of times each word occurred in each text sample. Each word therefore corresponds to a vector (a row of the matrix) of 37,651 elements, most of which of course equal zero.

Words that often occur together in a text sample will have vectors that are more similar to one another than the vectors of words that do not share many samples. However, there is more to a word's meaning than simply the contexts it does or does not occur in. Two words are also semantically similar if there is a third word they often share a context with. For example, one author may prefer to use *taxi* while another likes the word *cab* better. In that case, these two words will not very often occur in the same text sample, but since they are both likely to share many samples with words like *driver* and *street*, they are similar in meaning.

In order to find these higher-level relations, LSA uses a technique called singular value decomposition (for details, see the appendix of Landauer & Dumais, 1997), which replaces the matrix sketched above with one that has fewer, linearly independent, columns. Every row of the new vector still corresponds to a word, but the number of dimensions of these word vectors is much smaller than before. Landauer and Dumais (1997, p. 220) found that using approximately 300 dimensions resulted in the best performance on the synonym part of the Test of English as a Foreign Language. The same number of dimensions is used by the Predication model and for the results presented here.

The 300-dimensional space constructed by LSA is called the *semantic space* because words that are semantically related are represented by similar vectors in this space. The similarity of two vectors is defined as their normalized inner product, which equals the cosine of the angle between the vectors. The more related two words are, the more similar their vectors will be and the larger this cosine is. For instance, *taxi* and *cab* have a cosine of .57, while the cosine between *taxi* and *elephant* is only .03, indicating that these last two words are almost completely unrelated (the corresponding vectors are close to orthogonal). On average, the cosine between two word vectors is .02, with a standard deviation of .06. The LSA website offers the possibility to compare word vectors and find the words closest to any word in semantic space.

Another informative measure is the length of a word vector, which indicates how much LSA 'knows' about the word. In general, the more often a word occurs and the more specific its contexts, the longer its vector is. For instance, the vector length of the relatively infrequent word *taxi* is 0.26, while the more common word *street* has a vector length of 1.52. This relation between word frequency and vector length does not hold in general. Function words are very frequent but occur in all kinds of contexts. As a result, their meaning cannot be inferred from the contexts in which they occur, so they receive very short vectors. For example, the length of the vector for *the* is only 0.03.

LSA can also estimate the meaning of a collection of words, such as complete texts. The vector representing any collection of words is simply the sum of the individual word vectors. In summing, these vectors reinforce one another in some dimensions and cancel one another out in others. The idea is that the resulting sum vector will thereby catch the overall meaning of the collection, and not the individual word senses that are not relevant to the text. The meaning of the collection is investigated by looking at the cosine between the vector representing the collection and adequately chosen landmark vectors. The meaning of the collection 'is like' the landmark words that result in a large cosine with the collection's vector.

2.5.2 Predication

The Predication model takes LSA one step further by constructing vectors representing propositions from LSA's word vectors in semantic space. Take, for instance, the sentence *The horse ran*.[13] LSA creates vectors for collections of words by summing the individual word vectors. However, this summation does not take into account that different words play different roles in a sentence. In this case, *horse* is the noun phrase and *ran* the verb phrase, but this information is ignored by LSA.

The Predication model offers an alternative for creating sentence vectors.

[13] This example, like all others in this section, is taken from Kintsch (2001).

In theory, these vector representations are constructed by combining LSA with the Construction-Integration model. Be reminded that the Construction-Integration model assumes that a reader's world knowledge is stored in a network consisting of nodes that correspond to concepts and propositions. For Predication, the concepts are replaced by LSA's word vectors and the weights of the connections between concepts are set according to the cosines between the word vectors. The vector representing the proposition RAN(HORSE) is then claimed to be constructed as follows (Kintsch, 2001, pp. 179-181):

1. A network is constructed consisting of all 92,409 words in the semantic space. The connection weights between the *ran* node and all others equal the cosine between the corresponding vectors. The same applies to the connection weights between the *horse* node and the nodes 'close to' (i.e., having a large cosine with) *ran*. The weights between *horse* and the nodes not close to *ran* are set to negative values or to 0.
2. The other $\frac{1}{2} \times 92,407^2 \approx 4.3$ billion weights are set to a negative value, such that the sum of all weights equals 0. This ensures that only words related to both *horse* and *ran* will receive a positive activation value during the integration process.
3. The integration process from the Construction-Integration model (see Section 2.4.2) is applied to this network, resulting in an activation value for every node.
4. The vector representing RAN(HORSE) equals the sum of *all* 92,409 word vectors, weighted by their activation values.

Apart from practical issues when working with such a large network, it also poses a theoretical problem. It is likely that most words are hardly related to *ran* or *horse* at all. Therefore, most connection weights for these two nodes will be close to zero. Since all nodes other than *ran* and *horse* are negatively connected to each other, the many nodes unrelated to *ran* or *horse* will only have connections with negative weights. As was explained in Section 2.4, integration of a network that includes nodes with a negative weight sum is not guaranteed to be well defined: All activation values may end up at 0. In practice, however, this problem does not arise because a simplified model is applied instead of the theoretical model described above.

As a first simplification, not all words in the semantic space are added to the network but only the m words closest to *ran* are considered. The value of m varies greatly: from $m = 20$ in some simulations of Kintsch (2001), to $m = 500$ in Kintsch (2000) and Kintsch and Bowles (2002). Second, and most importantly, the integration process is not applied at all. Instead, the predicate (*ran*), the argument (*horse*), and the k words closest to the argument are assumed to receive a final activation of 1, while all others are 0. In Kintsch (2001), the values of k range from 1 to 5. Summing all word vectors, weighted by their activation as in step 4 above, now comes down to simply summing these k vectors, *ran*, and *horse*.

To summarize, consider a proposition of the form $P(A)$, with predicate P and a single argument A. Applying the two simplifications discussed above, the Predication model determines the vector representation of $P(A)$ as follows:

1. Take the m LSA vectors that have the largest cosine to P.
2. Of these, select the k vectors that have the largest cosine to A.
3. The vector for $P(A)$ equals the sum of these k vectors, P and A.

Note that if the cosine between A and P is large enough, P will turn up in the k vectors selected in step 2 and is included in the vector sum twice. For example, with $m = 20$ and $k = 5$, the vector representing RAN(HORSE) equals the sum of the predicate vector *ran*, the argument vector *horse*, and the $k = 5$ vectors *stopped*, *yell*, *came*, *saw*, and *ran* itself (Kintsch, 2001, p. 181).

To investigate the meaning of the resulting proposition vector, it has to be compared to several landmarks. For the sentences *The horse ran* and *The color ran*, Kintsch (2001, Table 2) uses as landmarks the vectors for *gallop* and *dissolve*. The first of these landmarks is closer to RAN(HORSE) while the second has a larger cosine with RAN(COLOR). It is concluded that the Predication algorithm correctly resulted in a vector for RAN(HORSE) that 'is like' *gallop*, and a vector for RAN(COLOR) that 'is like' *dissolve*.

2.5.3 Applications

Kintsch (2001) applies the Predication model to metaphor comprehension, causal inferencing, similarity judgement, and homonym comprehension. Since the first two of these applications are the most relevant to discourse comprehension, only these shall be discussed here.

Metaphor comprehension
One of the phenomena for which Predication offers an explanation is the comprehension of metaphor (Kintsch, 2000; Kintsch, 2001, pp. 186-189; Kintsch & Bowles, 2002). In a metaphor of the form *A is P*, the predicate *P* and the argument *A* can be quite unrelated. This also follows from the corresponding LSA vectors. Take for instance the metaphor *My lawyer is a shark*, corresponding to the proposition SHARK(LAWYER). The cosine between the LSA vectors representing *lawyer* and *shark* is $-.01$, showing the lack of relation between the two. This may cause difficulties when Predication tries to find words that are close to both the predicate and the argument in semantic space. In order to assure that some related words are found, the value of the parameter m needs to be increased. For metaphor comprehension, it is set to $m = 500$.

Apart from this alternative parameter setting, the Predication model for metaphor comprehension differs from other applications in one respect: There exists a threshold cosine below which word vectors are not included in the construction of the proposition vector (Kintsch, 2000, p. 259; Kintsch, 2001, p. 188). This threshold is set to two standard deviations above the average cosine between each pair of all LSA vectors, which equals $.02 + 2 \times .06 = .14$. The Predication algorithm for metaphor comprehension becomes:

1. Take the $m = 500$ LSA vectors that have the largest cosine to P.
2. Of these, select the $k = 5$ vectors that have the largest cosine to A.
3. Of these, reject the vectors whose cosine to P or to A is less than .14.
4. The vector for $P(A)$ equals the sum of A and the vectors resulting from step 3. Predicate P is included in the sum only if its cosine with A is larger than .14 (W. Kintsch, personal communication, October 9, 2000).

The vector representing SHARK(LAWYER) that results from this algorithm has a higher cosine with *vicious* (Kintsch, 2001, Figure 5) than does *lawyer* by itself, indicating that a shark-like lawyer is more vicious than just any lawyer. However, it is also more related to *fish*, which does not seem to be the intended meaning of the metaphor.

Causal inferences
Another application of the Predication model is the simulation of causal inferencing (Kintsch, 2001, pp. 189-192). A statement like *The student washed the table* implies that as a consequence *The table was clean*. If Predication picks up

this causal relation, the vector for WASHED(STUDENT,TABLE) should be closer to CLEAN(TABLE) than to, for instance, CLEAN(STUDENT).

Up to now, only vectors for propositions with a single argument have been constructed. For a proposition of the form $P(A_1, A_2)$, carrying arguments A_1 for the agent and A_2 for the patient, the Predication model does the following:

1. Take the $m = 20$ LSA vectors that have the largest cosine to P.
2. Of these, select the $k = 5$ vectors that have the largest cosine to A_2.
3. Take the $m = 20$ vectors that have the largest cosine to the vector formed by the sum of P, A_2, and the vectors selected in step 2.
4. Of these, select the $k = 5$ vectors that have the largest cosine to A_1.
5. The vector for $P(A_1, A_2)$ equals the sum of P, A_1, A_2, the k vectors from step 2, and the k vectors from step 4.

Using this algorithm, Kintsch (2001, Table 1) shows that WASHED(STUDENT, TABLE) indeed has a larger cosine with CLEAN(TABLE) than with CLEAN(STUDENT). More importantly, the difference between the two cosines is less when the sum of just the vectors *student*, *washed*, and *table* is chosen to represent *The student washed the table*. This is taken as evidence that, for causal inferences, the Predication model performs better than LSA's method of simply summing the vectors of the words involved.

2.5.4 Evaluation

Influence of vector length

In LSA, the vector representing any collection of words is simply the sum of the individual word vectors. The vector for $P(A)$ would be the sum of the vectors P and A. Predication biases this sum towards P by first selecting the m nearest neighbors of P and then selecting from these the k vectors closest to A, to include in the computation of the sum. As a result, the vector for $P(A)$ usually is closer to P than to A.

This bias can be overcome if the argument vector is much longer than the predicate vector, because the length of word vectors has a strong influence on the resulting proposition vector. When vectors are summed, the result is biased towards the longer vector. Be reminded that in LSA common content words generally have longer vectors than uncommon words. For instance, the word *bird* has a vector length of 2.04, while the length of *pelican* is only 0.15. This means that the impact *bird* has on a proposition vector is almost fourteen times larger than the impact of *pelican*. As a result, both BIRD(PELICAN) and PELICAN(BIRD) have vectors much closer to *bird* than to *pelican*. Kintsch (2001) argues that this asymmetry is in line with the way these propositions are usually verbalized:

> We say *The bird is a pelican*, providing information about some specific bird. Or we say *A pelican is a bird*, referring to the generic pelican. . . . The informationally empty *The pelican is a bird*, and the epistomologically [*sic*] empty *A bird is a pelican* are not common linguistic expressions. (p. 185)

However, the difference in expressing these propositions is not related to the relative vector lengths of *bird* and *pelican* but is caused by the fact that a pelican is a type of bird. For instance, the word *vertebrate* has a much shorter vector than *bird* (0.48 and 2.04, respectively) but we still say *A bird is a vertebrate* and *The vertebrate is a bird*, since a bird is a type of vertebrate.

Parameter sensitivity

A second problem is the model's oversensitivity to its parameter setting. The number of word vectors that are summed to compute the proposition vector is rather small ($k = 5$ at most). As a result, each of these k vectors (especially the long ones) can have a strong impact on the resulting vector for $P(A)$. Therefore, even the smallest change in the value of k or m can have large and unexpected consequences for the model's result.

For metaphor comprehension, the minimum cosine with P and A that a vector must have to be selected forms a third parameter. Changing this threshold value can cause a vector to suddenly appear in, or disappear from, the set of vectors whose sum equals $P(A)$, possibly resulting in a very different proposition vector. For instance, when constructing the vector for *My lawyer is a shark*, one of the $k = 5$ selected words is *caught*. With an accuracy of two decimals, the LSA website gives a cosine between *caught* and *lawyer* of .14, meaning that the exact cosine lies somewhere between .135 and .145, just above or just below the threshold. This makes quite a difference, since we found that the two possible vectors for SHARK(LAWYER) (either with or without *caught*) have a cosine of only .73 with each other.[14] The difference between the two vectors is more

[14] The other vectors that were included represented the words *lawyer*, *devilish*, *motored*, and *viciousness*.

serious than may seem at first. Without including *caught*, the cosine between SHARK(LAWYER) and *fish* is .06. When *caught* is included, this cosine increases to .33.

Metaphor comprehension

As was discussed above, the vector representing the proposition $P(A)$ is usually biased towards P. For metaphors, this would have the surprising result that *My lawyer is a shark* corresponds to a vector that is closer to *shark* than to *lawyer*. In other words, metaphors are taken literally. To counter this problem, the threshold cosine was introduced. If the cosine between the predicate and the argument is less than .14, the predicate vector is not included in the computation of the metaphor's vector. The cosine between *shark* and *lawyer* equals $-.01$, so in this case the shark-like lawyer is more of a lawyer than a shark. Also, the larger value of *m* used for metaphor comprehension helps in finding vectors close to the argument, thereby reducing the bias towards the predicate.

Cacciari and Glucksberg (1994) and Gibbs (1994) argue that comprehending a metaphor is no different from comprehending a literal statement. Kintsch agrees:

> Prior to that work, the dominant view was that the comprehension of nonliteral statements involves two steps: First, it must be recognized that the statement makes no sense if interpreted literally; then, its intended, nonliteral meaning is computed by some kind of inference. Now we know that, instead, metaphors can be understood directly, like literal statements. (Kintsch, 2000, p. 257)

Following this, he claims that the Predication model is indeed "a computational model of metaphor comprehension that treats metaphors in the same way as literal statements" (p. 257). However, this is clearly not true. First of all, a larger value of parameter *m* is needed for metaphor comprehension than for other applications of the Predication model. Second, introducing the threshold cosine for metaphor comprehension means that it is not even the same model. Therefore, modeling the comprehension of metaphor does involve two steps: First, the modeler recognizes that the statement is a metaphor and adjusts the value of *m* and the Predication model accordingly. Next, the metaphor's vector is constructed. Such a two-step process is exactly what Kintsch claims to avoid.

Causal inferences

LSA creates a semantic space in which words are close to one another if they often occur in similar contexts. If the Predication model can indeed construct *causal* relations from these *semantic* relations, that would be a spectacular result. Unfortunately, the model's results on causal inferencing are not convincing. All nine examples of causal inference in Kintsch (2001) can be shown to be based on the accidental choice of landmarks. For instance, the Predication model results in a vector for *The student washed the table* that is closer to *The table was clean* than to *The student was clean*, indicating that vectors representing causally related statements lie closer to one another in semantic space than vectors for causally unrelated statements. However, this result can be explained if we take into account that proposition vectors are biased towards the predicate (in this case, the verb *washed*). The last column of Table 2.6 shows that the cosine between the words *washed* and *table* (cos = .25) is much larger than between *washed* and *student* (cos = .02). Biasing the proposition vector towards the verb therefore results in a vector closer to *table* than to *student*, and closer to CLEAN(TABLE) (cos = .83) than to CLEAN(STUDENT) (cos = .62). If the landmarks *The table was tired* and *The student was tired* are chosen to test causal inferences, the model incorrectly concludes that washing the table is more tiring for the table than it is for the student: We found a cosine between WASHED(STUDENT,TABLE) and landmark TIRED(STUDENT) of .66, compared to a cosine with landmark TIRED(TABLE) of .81.

Table 2.6: Cosines between four test propositions and landmark propositions, according to Kintsch (2001, Table 3); and cosines between the test proposition's predicate and the landmarks' arguments, according to the LSA website at lsa.colorado.edu.

proposition (pr)	landmarks (lm)	cos(pr, lm)	cos$(P_{\text{pr}}, A_{\text{lm}})$
WASHED(STUDENT,TABLE)	CLEAN(STUDENT)	.62	.02
	CLEAN(TABLE)	**.83**	**.25**
DROPPED(STUDENT,GLASS)	BROKEN(STUDENT)	.87	.10
	BROKEN(GLASS)	**.91**	**.23**
DRANK(DOCTOR,WATER)	THIRSTY(DOCTOR)	**.83**	.05
	THIRSTY(WATER)	.78	**.26**
SHOT(HUNTER,ELK)	DEAD(HUNTER)	**.73**	**.41**
	DEAD(ELK)	.70	.24

In three out of four cases in Table 2.6, the landmark closest to the test proposition was the one whose argument was closest to the proposition's predicate.[15] Only in the example DRANK(DOCTOR,WATER) did this not occur: Even though *drank* is closer to *water* than to *doctor*, the model concluded correctly that it was the doctor, and not the water, who must have been thirsty. However, changing the landmarks shows that the model is not able to find causal relations in general. Drinking the water will refresh the doctor and not the water, but we found a much smaller cosine of DRANK(DOCTOR,WATER) with REFRESHED(DOCTOR) (cos = .38) than with REFRESHED(WATER) (cos = .69).[16]

Kintsch (2001, p. 192) gave five more examples of causal inference by the Predication model. In each of these examples, shown in Table 2.7, there was only one landmark (e.g., *The pain went away*), which was compared to two test propositions varying only in the predicate. One of these formed a direct cause for the landmark (e.g., *Sarah took the aspirin*) and the other did not (e.g., *Sarah found the aspirin*). In all five cases, Kintsch found that the causally related proposition was closer to the landmark than the non-related proposition. However, we found that all these cases could be explained by simply looking at the cosine between predicates of the test propositions and of the landmark (Table 2.7). The landmark's predicate is always closer to the predicate of the causally related test proposition than to the other predicate. For instance, *went* is closer to *took* (cos = .74) than to *found* (cos = .39). Since a proposition's vector is usually closest to its predicate, this explains why these five landmark propositions are closer to the causally related propositions than to less related propositions.

2.5.5 Conclusion

LSA vectors have shown useful for semantically representing words. When it comes to sentences, however, its simple summing rule for combining word vectors is not powerful enough because it does not take syntactic information

[15] Note that the Predication model does not make the correct causal inference in the last example: The vector representing *The hunter shot the elk* is closer to *The hunter is dead* than to *The elk is dead*.

[16] In constructing the vector for REFRESHED(DOCTOR), LSA came up with the following $k = 5$ words: *ocassion* [*sic*], *hogarthian*, *gethsemane*, *chinoiserie*, and *carissima*. All of these occur only once in the corpus that was used to construct the semantic space, so their vectors have an extremely short length (0.03). Replacing them with the next five, more acceptable, words *sleeping*, *awake*, *soundly*, *evening*, and *sleep*, resulted in a vector for REFRESHED(DOCTOR) that had a cosine of .42 with DRANK(DOCTOR,WATER).

Table 2.7: Cosines between five landmark propositions and test propositions, according to Kintsch (2001, p. 192); and cosines between the predicates of the test propositions and the landmark, according to the LSA website at lsa.colorado.edu.

test propositions (pr)	landmark (lm)	cos(pr, lm)	cos(P_{pr}, P_{lm})
TOOK(SARAH,ASPIRIN)	WENT(PAIN,AWAY)	**.89**	**.74**
FOUND(SARAH,ASPIRIN)		.47	.39
EXPLODED(HARRY,PAPER BAG)	JUMPED(HE,in: ALARM)	**.32**	**.36**
INFLATED(HARRY,PAPER BAG)		.28	.07
SHOT(HIKER,DEER)	DIED(DEER)	**.74**	**.34**
AIMED(HIKER,at: DEER)		.56	.18
SCRUBBED(TED,POT)	SHONE(POT,BRIGHTLY)	**.45**	**.25**
FOUND(TED,POT)		.41	.19
LOST(CAMPER,KNIFE)	SAD(CAMPER)	**.48**	**.37**
DROPPED(CAMPER,KNIFE)		.37	.27

like word order into account. The Predication model solves this by biasing the result towards the vector that represents the predicate.

The model suffers from several technical problems, such as oversensitivity to parameter values and vector lengths. A more reliable algorithm for asymmetrically combining predicate and argument vectors can possibly result in a useful model for simulating, for instance, metaphor comprehension. For causal inference, however, this idea is not likely to work. The reason is that there is an important difference between words and propositions that the Predication model seems to ignore: Unlike words, propositions are statements to which a truth value can be assigned. Causal inferencing is based on these truth values and therefore requires propositions: If *it is true* that Sarah took the aspirin, then *it is likely* that the pain will go away.

LSA does not result in representations for propositions, but only for (collections of) words. Does the Predication model actually create proposition vectors from these word vectors? Kintsch introduces the model as a way to construct context-sensitive predicate vectors, indicating that the model does not create vectors for full propositions: "In N-VP sentences, the precise meaning of the verb phrase depends on the noun it is combined with. An algorithm is described to adjust the meaning of a predicate as it is applied to different arguments" (Kintsch, 2001, Abstract). On several later occasions, however, he does refer to vectors that represent propositions: "Consider a proposition of the form

P(A), where P and A are terms in the LSA semantic space represented by vectors. In order to compute the vector for P(A) . . . " (p. 179).

This latter view on the nature of Predication vectors is the less accurate one. LSA constructs a semantic space in which the cosine between vectors is a measure for the semantic relatedness between the corresponding words. Predication takes the LSA vectors and constructs from them other vectors *in the same space*. Relations that depend on truth values (such as causal relations) do not correspond to any measure in the semantic space. This is easily shown by comparing the vectors for CLEAN(TABLE) and its negation NOT(CLEAN(TABLE)). These vectors are very similar (their cosine is a very high .91), showing that they do not represent the opposite meanings of the propositions. Instead, Predication vectors correspond to words adjusted to a specific context. The vector that is claimed to represent the sentence *The horse runs* should actually be interpreted as meaning something like *horselike running*, since it takes the verb *runs* and adjusts it to the context of *horse*.

It is certainly possible to create a space in which vectors correspond to propositions, but this cannot be the same space that words reside in. The following section describes a model that, after sufficient training, does construct vectors that represent propositions.

2.6 The Gestalt models

Although a proposition can be expressed as a sequence of words (in fact, people do this all the time), simply combining representations of words does not result in a representation of the proposition expressed by these words, at least, not if propositions are regarded as statements carrying truth values. A semantic representation of propositions will always need to include knowledge about probabilistic relations among the propositions. In the previous section, it was argued that for this reason the Predication model cannot be said to construct representations of propositions.

In a localist model, it is possible for the modeler to estimate a connection weight between any two items, thereby constructing a representation based on the modeler's world knowledge. As discussed in Section 1.2.1, however, a *distributed* model may be more suitable for simulating discourse comprehension. Since finding distributed representations requires the fine-tuning of many parameters that do not have any clear meaning and are therefore almost impossible to estimate directly, distributed representations are usually computed on the basis of a large corpus rather than hand coded. Latent Semantic Analysis is an example of this, since its word vectors are computed from a large number of naturally occurring texts.

Just like word vectors can be extracted from a corpus of words, proposition vectors should be based on a corpus of propositions. However, two problems immediately arise. First, there neither exists a naturally occurring corpus of propositions, nor is there a system that can automatically parse a text into propositions. Second, even if a corpus of propositions is available, the number of possible propositions will be much larger than the number of proposition in the corpus because new propositions can be composed from known concepts and propositions. A proposition that never occurred in the corpus may easily show up in a text that is to be processed by the model.

A solution to the first problem is to use an artificial corpus, covering only a small domain known as a *microworld*. Of course, a model that has developed distributed representations of microworld propositions cannot be used to process propositions that deal with events outside the microworld. The model only has knowledge of the microworld, which is at best a subset of an actual reader's world knowledge. Consequently, a detailed, quantitative comparison of simulation results and empirical data is not possible. However, qualitative comparisons can still indicate whether or not the model adequately simulates human discourse comprehension processes.

If the corpus contains all possible propositions a sufficient number of times, a representation for each of them can be found. The second problem is then not so much solved as it is avoided. On the other hand, if the corpus does not contain all possible propositions, the model has to be able to adequately represent propositions it has never seen before. In short, the model has to generalize from known examples. Whether this is possible in practice remains to be seen.

In this section, we discuss two strongly related models that develop distributed representations of (sets of) propositions from an artificial corpus describing a microworld. The Sentence Gestalt model (St. John & McClelland, 1990, 1992) takes a sentence as input and develops a representation for the proposition described by the sentence. The Story Gestalt model (St. John, 1992; St. John & McClelland, 1992) takes a sequence of propositions as input and develops a representation for the complete story formed by the propositions.

2.6.1 The Sentence Gestalt model

St. John and McClelland (1990, p. 222) give the sentence *The pitcher threw the ball* as an example in which all content words are ambiguous. The word *pitcher* can refer to either a ball player or a container for liquids; *threw* means either *tossed* or *hosted*; and a *ball* can be a sphere or a party. The complete sentence is ambiguous as well: The pitcher either tossed a sphere or hosted a party, but considering the most common uses of *threw* and *ball*, and the baseball sense of *pitcher*, the latter interpretation is less likely.

The Sentence Gestalt model is able to take as input such a sentence, with all its ambiguities, and to construct from it a distributed representation of the situation described by the sentence. As much as possible, ambiguities are resolved and missing information is filled in. The model accomplishes this by finding regularities in a corpus of example sentences and corresponding events. For example, it learns that the sort of *ball* thrown by a pitcher is usually the one that can be hit with a bat, and that a *pitcher* who throws balls is not the sort of pitcher commonly found in the kitchen.

The corpus

The corpus consists of sentence/event pairs. The event refers to something that can be observed in the world, while the sentence is a way to describe (part of) this event linguistically. The model extracts the meaning of words and sentences from regularities in the corpus. For instance, the word *pitcher* occurs in sentences that describe events in which the concept PITCHER-PERSON plays a role, but also in events that include a PITCHER-CONTAINER. The word *pitcher* is therefore inferred to refer to both these concepts.

In the microworld (as in reality) it is much more common for a PITCHER-PERSON to toss a sphere than to host a party. As a result, the model infers the more likely meaning of the ambiguous sentence *The pitcher threw a ball.* Since a PITCHER-CONTAINER never tosses spheres or hosts parties, and parties do not get tossed nor are spheres ever hosted, all these grammatically correct but senseless readings of the sentence are ignored.

Fifty-two different words are used in the corpus sentences, including words that can be either noun or verb (e.g., *rose*), pronouns, prepositions, and adverbs. To make passive sentences possible, the auxiliary verb *was* and preposition *by* are included as well. Articles (*the* and *a*) are not used and verbs are not inflected. As an example, the English sentence *The ball was thrown by the pitcher in the park* corresponds to *ball was threw by pitcher in park* in the corpus.

A sentence rarely specifies a unique event. This is not only because of ambiguity, but also because events are usually not described in full. For instance, the event in which the pitcher threw the ball to the schoolgirl in the park can be expressed by the sentence *ball was threw by pitcher*, ignoring the park and the schoolgirl. In the corpus, every sentence is paired with an unambiguous, fully instantiated event that is (partly) expressed by the sentence. There are 120 different events possible in the microworld, but because of the many possible ways to describe them, the number of sentence/event pairs is 22,645 (St.John & McClelland, 1990, p. 227).

Constructing representations

Sentences are converted into proposition vectors by a recurrent neural network of which Figure 2.8 shows the architecture. Recurrent networks are similar to feedforward networks, except that they have a set of untrainable connections, linking one of the intermediate layers to itself or to an earlier layer. As a result, the activation pattern in the hidden layer forms part of the next input that is to be processed. Recurrent networks are commonly used to find patterns in temporal sequences, such as phonemes in a word, or words in a sentence (Elman, 1990, 1993).

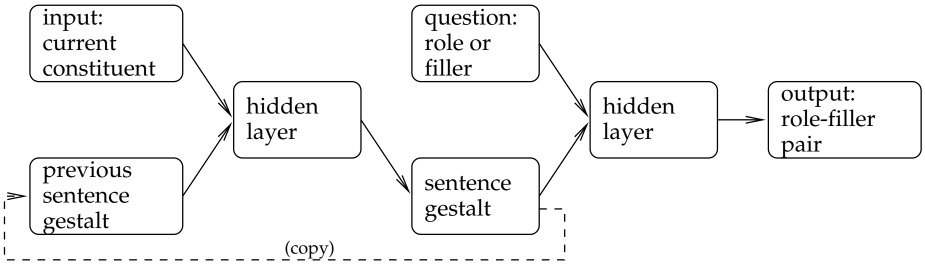

Figure 2.8: Architecture of the Sentence Gestalt network. Each block is a layer of neural network units. The 'question' and 'output' layers each consist of 68 units. The 'input' layer has 56 units. All other layers have 100 units. Solid arrows indicate that each unit in a layer is connected to all units in the next layer. The dashed arrow indicates that the activation vector in the 'sentence gestalt' layer is copied to the 'previous sentence gestalt' layer.

First, the sentence is divided into sentence constituents of one or two words. For instance, the word sequence *ball was threw by pitcher in park* consists of four constituents: *ball*, *was threw*, *by pitcher*, and *in park*, which enter the network one by one. The network's input layer contains a unit for each word, so it represents words locally. Additionally, it has four units that represent the position of the constituent in the sentence: 'preverbal' (the first constituent), 'verbal' (second), 'first postverbal' (third), or 'more postverbal' (fourth or more).

When the first constituent *ball* enters the network, only the input units labeled 'preverbal' and 'ball' are activated. This activation pattern is propagated through the hidden layer to the layer marked 'sentence gestalt'. The pattern of activation over its units forms the vector representation of the sentence so far. This activation pattern is copied to the 'previous sentence gestalt' layer. Now, the next sentence constituent enters the network. The input layer activates the units 'verbal', 'was', and 'threw', and these activations are fed into the hidden layer, where they are combined with the activation pattern coming from the 'previous sentence gestalt' layer. The result is passed on to the 'sentence gestalt' layer, which should now represent the incomplete and ambiguous statement *The ball was thrown*.

This process is repeated until all four sentence constituents are processed. The sentence gestalt layer should now represent something like the proposition THREW-TOSSED(PITCHER-PERSON,BALL-SPHERE,location:PARK), although the representation may also correspond to the alternative, much less likely reading THREW-HOSTED(PITCHER-PERSON,BALL-PARTY,location:PARK).[17]

Training the network

Of course, claiming that a proposition is represented does not mean that the representation is in any way adequate or useful. To test for this, the network can be probed by activating one of the units in the 'question' layer. Each of the question units corresponds to a concept (like THREW-TOSSED, PITCHER-PERSON, or PARK) or to one of nine roles for which the concepts are possible fillers (like 'action', 'agent', and 'location'). Activating the 'action' unit comes down to asking what action is performed in the proposition. If the network performs correctly, the output layer, which also consists of a unit for each role and for each filler, answers by activating the 'action' and THREW-TOSSED units.

If the network does not perform correctly, the difference between the correct answer and the given answer can be used to update the connection weights according to the well-known backpropagation algorithm (see e.g. Rumelhart, Hinton, & Williams, 1986). As a result, the vector representations of propositions become more suitable for the task of answering questions. After being trained in this way on over 300,000 sentence/event pairs, the network answers correctly in over 99% of the cases (St.John & McClelland, 1990, p. 230).

2.6.2 The Story Gestalt model

The Story Gestalt model works very much like the Sentence Gestalt model. Just like a sequence of words is a (possibly incomplete) description of an event, a sequence of propositions is a (possibly incomplete) description of a story. The Story Gestalt model constructs a distributed representation of the story described by the propositions, using regularities in a corpus of stories to fill in missing information.

[17] All verbs in the corpus sentences are in past tense form, so a noun like *park* is not ambiguous.

The corpus

In the Sentence Gestalt model, a distinction was made between a sentence and the events it describes. A similar distinction between *proposition sequences* and *event sequences* is made for the Story Gestalt model. The corpus consists of temporal sequences of propositions ('stories'), describing sequences of events. A typical example is the story in Table 2.8, consisting of four propositions.

Table 2.8: An example story consisting of four propositions, and a corresponding story text.

t	proposition	possible text
1	DECIDED-TO-GO(ANDREW,BAR)	*Andrew decided to go to the bar.*
2	MADE(ANDREW,SARAH,PASS,OBNOXIOUS)	*He made an obnoxious pass at Sarah,*
3	GAVE(SARAH,ANDREW,SLAP)	*who gave him a slap.*
4	RUBBED(ANDREW,CHEEK)	*Andrew rubbed his cheek.*

Event sequences are always complete and unambiguous: Everything that happens is stated in the event sequence and every event is a unique combination of predicate, argument, patient, etcetera. The total number of event sequences is 28,480, but the number of possible proposition sequences is much larger because character's names can be replaced by pronouns or a complete proposition may be deleted from the sequence.

Constructing representations

Architecturally, the two Gestalt models are identical, as can be seen from Figure 2.9. Input and output representations are localist: There is one unit for each concept. Concepts that can occur in different roles have a unit for each possible role. For instance, there is a unit for ANDREW-AGENT, for ANDREW-PATIENT, and for ANDREW-RECIPIENT, making it possible to tell the difference between *Andrew made a pass at Sarah* and *Sarah made a pass at Andrew*.

After processing a proposition, the activation pattern in the 'story gestalt' layer represents the story so far. This activation vector is copied to the 'previous story gestalt' layer, where it serves as additional input when the next proposition is processed.

Training the network

Each unit of the 'question' layer represents a verb. The adequacy of the distributed representation in the story gestalt layer is tested by activating one of

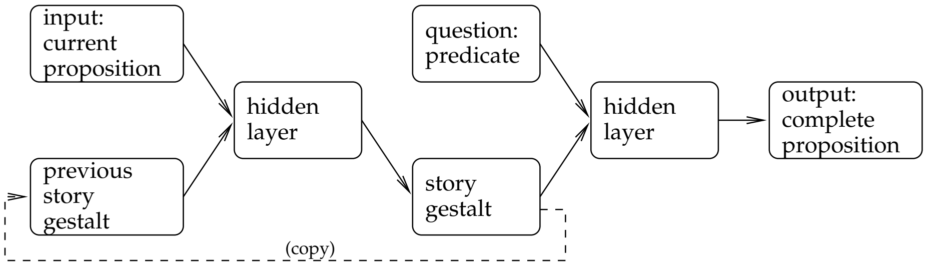

Figure 2.9: Architecture of the Story Gestalt network. Each block is a layer of neural network units. The 'input' and 'output' layers each consist of 136 units. The 'question' layer has 34 units. All other layers have 100 units. Solid arrows indicate that each unit in a layer is connected to all units in the next layer. The dashed arrow indicates that the activation vector in the 'story gestalt' layer is copied to the 'previous story gestalt' layer.

these units. For instance, activating the GAVE unit corresponds to asking the question: 'Who gave what to whom and where?'. If the story of Table 2.8 is processed by a sufficiently trained network, only the output units GAVE, SARAH-AGENT, ANDREW-RECIPIENT, SLAP, and BAR should become active in response to the question GAVE.

If the network does not answer correctly, the backpropagation algorithm can be used to update the connection weights. All 28,480 event sequences are used for this training process. For each of the event sequences, there exists a unique proposition sequence that fully and unambiguously describes the event sequence. Before this proposition sequence is used as training input, the character's names can randomly be replaced by pronouns, and propositions can be removed from the sequence. The event sequence, however, remains intact.

After processing each proposition, the network has to answer questions regarding the whole story, including the propositions not yet processed. For instance, after processing DECIDED-TO-GO(ANDREW,BAR), the network already has to predict what the arguments of the RUBBED predicate will be. In this way, the network is trained to predict future story events from the story processed so far. Also, when answering a question, the network is not allowed to respond with a pronoun but has to fill in the character's name.

The difference between the network's output and the correct answer to the question is backpropagated to update the connection weights. After being trained on one million sequences of propositions, the network can accurately resolve pronouns (St.John, 1992, pp. 279-280).

2.6.3 Evaluation

The Sentence Gestalt model

In Section 2.5.5, we claimed that the vectors resulting from the Predication model do not represent propositions because they do not correspond to statements about events in the world. The Sentence Gestalt model, on the other hand, does represent propositions, as can be seen from its ability to make inferences. For instance, if it is stated that *The teacher ate the soup*, it is very likely that a spoon was used. This does not follow from the semantics of the words *teacher*, *ate*, and *soup*, but from world knowledge about the instrument that is commonly used when soup is eaten. When this sentence is processed by the Sentence Gestalt model, it constructs a propositional representation that is more than simply a combination of the words. If the network is asked what instrument is used in the proposition, it answers SPOON (St. John & McClelland, 1990, p. 234).

Ambiguous words and sentences are correctly interpreted. When processing *The pitcher hit the bat with the bat*, the model correctly infers that *pitcher* is a PITCHER-PERSON rather than a PITCHER-CONTAINER, and that the second *bat* is a BAT-BASEBALL rather than a BAT-ANIMAL. The first *bat* is more likely to be an animal, but it cannot be excluded that it is a baseball bat. The model responds accordingly when asked about the patient of the sentence: The BAT-ANIMAL output unit is activated more strongly than the BAT-BASEBALL unit (St. John & McClelland, 1990, pp. 232-233).

Can the model represent propositions it has never seen before? The corpus described above contains all sentence/event pairs possible in the microworld, so the model's ability to generalize to new sentences cannot be tested. Therefore, St. John & McClelland (1990) constructed a second microworld, consisting of 10 people and 10 reversible actions. Sentences can appear in active or passive voice, making a total of 10 (people) × 10 (actions) × 10 (people) × 2 (voices) = 2,000 possible sentences of the form *John saw Mary*. Of these, 250 are not included in the corpus. After learning to represent the other 1,750 sentences, the model can also correctly process 97% of the 250 unseen sentences (St. John & McClelland, 1990, p. 242). This shows that it is able to use regularities in the corpus to interpret new sentences.

The Story Gestalt model

The Story Gestalt model, too, is able to pick up regularities in the training corpus, and use these to make inferences about story events. For instance, stories taking place in a bar always end in a man rubbing his cheek after being slapped by a woman, or the man rubbing lipstick after being kissed by her. Whether she is more likely to slap or to kiss him depends on the kind of pass he makes. An obnoxious pass results in a slap in 70% of the cases, while 70% of the polite passes result in kissing. Since half of the passes in the corpus are obnoxious and half are polite, the a priori probabilities of rubbing cheek and of rubbing lipstick are both .5. In accordance with this, after processing only a proposition about someone going to a bar, the network answers with a small activation of both the CHEEK and LIPSTICK output units when asked the arguments of RUBBED (St. John, 1992, Table 4).

When it is known what kind of pass is made, it is easier to predict whether a cheek or lipstick will be rubbed at the end of the story. An obnoxious pass results in a probability of .7 for CHEEK and of .3 for LIPSTICK. After a polite pass, these values are reversed. Again, the network's output activations lie very close to these values.

After the story has stated whether the seducer gets slapped or kissed, it can be predicted with certainty whether he will rub his cheek or lipstick. The model now strongly activates the correct output unit, even if the other one was more active previously. This shows the model's ability to make predictive inferences, and to revise them if new information supports an alternative inference.

The model can also make correct inferences that contribute to the story's coherence. Stories taking place in restaurants, for instance, always involve something being ordered. Moreover, the person who orders is always the one who pays the bill later. When the model processes the story *Albert and Clement decided to go to a restaurant. The restaurant was expensive. Clement paid the bill*, the model correctly infers the missing information that Clement ordered (St. John, 1992, Table 3).

As a result of the extreme regularities in the corpus stories, generalization is rather poor. For instance, the network learns that in expensive restaurants people always pay with credit card but in cheap restaurants the bill is payed in cash. After processing a new story about an expensive restaurant where the bill is payed in cash, the model incorrectly answers CREDIT-CARD to the question how the bill was payed. In the corpus, the association between EXPENSIVE

RESTAURANT and CREDIT-CARD is so strong that the contradictory input is overruled. It must be noted here that the model only knows about the microworld from the corpus stories it is trained on. Since in none of these stories cash is used to pay in expensive restaurants, paying cash in an expensive restaurant is just as impossible to the model as is rubbing lipstick after being slapped.

Generalization is possible if the microworld is somewhat more complex. In a second simulation, St. John (1992) added eight locations where bills are paid. Each location was either cheap or expensive, but this did not correlate with the method of payment: Cash and credit card were both accepted. When it came to restaurants, however, the bill was always payed by credit card in expensive restaurants, and with cash in cheap restaurants. After learning about this new microworld, the model did show generalization: Stories in which cash was used in expensive restaurants, or credit cards in cheap ones, were processed correctly even though they had not been seen before (St. John, 1992, Table 5).

2.6.4 Conclusion

The Gestalt models' recurrent nets, like all neural networks, adapt to the task they are to perform, which in this case is to answer questions about the sentence or story that was processed. For the Sentence Gestalt model, this comes down to finding the most likely filler of a given role (or vice versa) in the sentence. For the Story Gestalt model, the questions are predicates from the story and the network is trained to reproduce or infer the predicate's arguments.

The distributed representations of sentences or stories developed by the networks during training do not encode knowledge in general because they serve no other purpose than to be useful for performing the tasks on which the networks are trained. Consequently, they are not likely to be useful for anything else. In LSA, vectors for individual words could be summed to create a vector representing the collection of these words. Vector representations in the Gestalt models, on the other hand, cannot be manipulated so easily. They can only be used by the trained network that probes their meaning by asking a question. It can therefore not be said that the representations carry any meaning independent from the network that constructed them. If they are to be used for anything else (i.e., summarizing a story) a new network needs to be trained to perform the new task. However, since the vector representations were developed for another purpose, they may not be up to this new task. In short, it is questionable to what extent a vector carries 'the meaning' of a sentence or story in general. Because of this, the model has limited generality concerning the tasks it can perform.

The Sentence Gestalt model learns to complete any role/filler combination that applies to a sentence, which is likely to result in a fairly general representation of the meaning of the sentence. The questions the Story Gestalt model learns to answer, on the other hand, are quite limited, possibly resulting in more task-specific representations than the ones developed by the Sentence Gestalt model. For instance, the Story Gestalt network is never asked about the temporal order in which the story's events occur, so this order is not represented. As a result, the model does not learn the notion of *story time*. For sufficient comprehension, of course, this knowledge is crucial. It cannot be hard to teach the network to answer questions like 'what happened before Clement paid the bill?' because story events always happen in the same order. However, in a slightly more complex microworld where it is, for instance, possible to pay the bill *before* eating the food (as one might do when in a hurry), learning to represent story time may not be so easy. It is unclear to what extent the model can accomplish this.

Apart from story time, both Gestalt models lack a notion of *processing time*. Processing a piece of input, or answering a question, always takes one sweep through the network. Therefore, the models cannot make predictions about reading times or response times, making them considerably less descriptively adequate. It has even been argued (see e.g. Van Gelder & Port, 1995) that psychological processes cannot be modeled without including a notion of the time in which these processes take place. In that case, the Gestalt models should not be considered psychological models at all.

The questions posed in the beginning of this section were whether it is possible to represent propositions distributively, and whether such a representation can be constructed for previously unseen propositions. The Sentence Gestalt model shows that both answers are yes. Moreover, the Story Gestalt model shows that sets of propositions can be represented distributively too, which is an important step towards a distributed model of discourse comprehension.

An interesting subject for further investigation is the possibility of coupling the two Gestalt models. In principle, the representations of propositions developed by the Sentence Gestalt model could be used as input to the Story Gestalt model, instead of its localist input. Miikkulainen and Dyer's (1991) DISPAR model (see also Miikkulainen, 1993) can be viewed as a combination of the Gestalt models. It takes as input microworld stories in the form of sequences of words and is trained to paraphrase the stories. During training, it develops distributed representations of both propositions and stories. In contrast to the Gestalt models, DISPAR also develops distributed representations of the words that form its input. The model's drawbacks, however, are similar to those of the Gestalt models: DISPAR does not have a notion of processing time either, and it can only reproduce the temporal order of story events because this order is fixed.

In the next chapter, a model is discussed in which the order of story events is represented explicitly. Moreover, it does allow for predictions of reading time.

2.7 Conclusion

As was noted in the introduction to this chapter, a direct comparison among the seven models discussed here is not possible because of the variety in the modeled aspects of discourse comprehension. It is possible, however, to judge the quality of the individual models using four of Jacobs and Grainger's (1994) criteria for model evaluation: simplicity, descriptive adequacy, explanatory adequacy, and generality. For each of these criteria, we shall discuss only those models that do either much better or much worse than the others.

Simplicity
A simple model has few free parameters, which are preferably psychologically interpretable. Also, it should have few architectural assumptions not meant to simplify the model. The Landscape model is a good example of simplicity: It has no parameters and the rules for computing concept (association) strength are clear and make sense intuitively. On the other extreme, the Construction-Integration model's integration process is defined by three equations (spreading activation, discarding negative values, and normalization) that seem quite ad hoc and do not have a clear rationale: Negative values need to be discarded to make normalization possible, which is needed to prevent activation values from rising unlimitedly. However, a slightly more sophisticated equation for activation spreading could achieve the same. Also, connection weights are set by hand for every task and input text, constituting many free parameters. In comparison, the Gestalt models contain far more connection weights but since these are computed by a training process over a set of realistic examples and they are the same for any input, they cannot be considered free parameters.

Descriptive adequacy
A model is descriptively adequate if it predicts empirical findings. Of the seven models in this chapter, the output of the Langston and Trabasso model is validated against the largest data set, but this was shown to be hardly the model's own merit. Similarly, recall data was predicted by the Landscape model, but this was mostly due to adequate input activations and not the result of the model's simple design. Contrary to this, the Resonance model was not validated directly against empirical data, even though the model was designed in

order to explain experimental results. The Gestalt models, too, did not have their results compared to any data.

Explanatory adequacy
Models should not only produce results that are consistent with empirical data but also offer an explanation of these results. The Langston and Trabasso model, for instance, does not have any explanatory adequacy since it merely states the well-known fact that causal relations in a story predict many empirical observations without explaining *why* they do so or *how* such causal relations are found by the reader. Similarly, the Landscape model only formalizes the idea that simultaneous activations in working memory lead to association in episodic memory.

The Construction-Integration model claims to explain how relevant world knowledge becomes part of a textbase but, as we have seen, this explanation does not suffice. The Predication model forms an explanation of the context-sensitive interpretation of words, but not of the comprehension of propositions. The only models that have any explanatory adequacy are the Resonance model and the Gestalt models. The Resonance model explains how a bottom-up process can cause reinstatement of text items, provided that world knowledge is not required. The Sentence and Story Gestalt models explain how the interpretation of sentences and stories depends on world knowledge.

Generality
A general model is one that can be applied to several inputs and can perform different tasks. As far as input is concerned, the Resonance model may not be very general because a suitable setting of its parameters seems quite text-specific. This problem is even worse in the Construction-Integration model, for which 'general' knowledge has to be implemented for each input text individually. All other models can easily process different inputs, although preparing these inputs for the Landscape model and the Langston and Trabasso model requires quite a lot of effort from the modeler. On the other hand, the Predication model can be applied directly to any set of LSA vectors, the Sentence Gestalt model can process any sentence that consists of known words, and the Story Gestalt model can process any story as long as it takes place within the microworld on which is was trained.

As for task generality, the Construction-Integration model has been applied to many sorts of tasks (as well as texts), although it must be added that this may not result from its generality but from the fact that it is simply the oldest and best known of the seven models. Also, each of these tasks required its own setting of connection weights, decreasing the model's task generality. The Landscape model only performs the task of computing the memory strength of (associations between) text items, so it does not score well on task generality. The Langston and Trabasso model is even worse, however, since it can be argued that it does nothing whatsoever. The Gestalt models, as we have seen, can only perform the tasks on which they were trained but, at least in theory, they can be trained on several tasks. Only the Predication model has been used for a variety of applications, but some of these did require a change in parameter setting.

Computational soundness

Jacobs and Grainger (1994) not only discuss computational models but also apply the above criteria to verbal models, which are not formalized precisely enough to be implemented as a running simulation. Consequently, they leave out the most important property for computational models: *computational soundness*. A computational model should have some important properties regarding stability: Values should not increase or decrease unlimitedly; The relation between model values and psychological measures should not change over tasks or inputs, let alone during a simulation; If the process does not converge, it should not be oscillating between senseless states; Small changes in parameter values should not generally result in sudden and implausible changes in output or model behavior, unless such a 'phase transition' is the cognitive phenomenon being modeled (see e.g. Pollack, 1995).

Another reason why Jacobs and Grainger do not note this important criterion might be that they focus specifically on modeling visual word recognition, a field of research much further developed than modeling discourse comprehension. For word recognition models, computational soundness may be so much taken for granted that it does not even need to be mentioned. For models of discourse comprehension, the situation is quite different. In fact, five out of the seven models from this chapter cannot be said to be computationally sound. The Resonance model, the Landscape model, and the Langston and Trabasso model all contain values that can rise unlimitedly. Especially in the Resonance model values rise dramatically. The Construction-Integration model may end up oscillating between senseless states, and it has been shown oversensitive to the setting of its connection weights. The Predication model, too, turned out to be oversensitive to its parameter values. Only the Gestalt models can be said to be computationally sound, which is not surprising since they are based on well established neural network technology. Another model that was built upon sound mathematical foundations is presented in the following chapter.

3

The Golden and Rumelhart model

A was mentioned several times before, one of the most important processes during discourse comprehension is the activation of the reader's general knowledge, which can lead to inference of relevant information not explicitly mentioned in the text. Out of the seven models discussed in the previous chapter, only two simulate this inference process: the Construction-Integration model and the Story Gestalt model. These models differ strongly in how they view inferencing. In the construction phase of the Construction-Integration model, individual text propositions retrieve a set of associated propositions from the reader's world knowledge net. During the following integration phase, the propositions that are most appropriate to the text are selected. In this view, inference is the result of a search process through the reader's world knowledge. The story text forms the starting point for this search. As discussed in Section 2.4.4, one of the main problems with the Construction-Integration model is the subjectivity involved in defining the part of the world knowledge net that is included in the model.

The Story Gestalt model offers an alternative view. According to that model, world knowledge is formed by accumulated experiences of event sequences in a microworld. A story is considered to be an incomplete description of a sequence of events, which can be completed if it matches previously seen patterns. Inferring propositions comes down to filling the gaps in the story by a process of pattern completion.

The Story Gestalt model suffers from two major shortcomings. First, the order of story events is not represented, so the model has no notion of story time. Second, processing a story proposition always requires the same number of computations, so there is no notion of processing time either. Both these problems are solved by the model proposed by Golden and Rumelhart (1993; Golden, Rumelhart, Strickland, & Ting, 1994), discussed in detail in this chapter. It is similar to the Story Gestalt model in the sense that it also views inference as a form of pattern completion. However, it does involve a switch back from distributed to localist representations.

3.1 Model architecture

3.1.1 Trajectories through situation space

At any moment in a story, a reader may believe one or more propositions to be the case in that story, either because the story text mentioned the proposition or because the reader inferred it from other propositions. The collection of propositions believed to be the case at one moment in the story is called the *story situation* at that moment. In the Golden and Rumelhart model, these propositions are taken from a set of n different propositions, which includes at least those stated in the story text and those that need to be inferred for adequate comprehension of the story.

The temporal order in which story situations occur is represented explicitly by associating a *story time step* index $t = 1, 2, 3, \ldots$ to every situation. A situation's time step index denotes its position in the story's chain of events. The situation at time step $t - 1$ occurs before (and is a possible cause of) the situation at t. Likewise, the situation at $t + 1$ is a possible consequence of the situation at t. Note that time steps refer to the temporal order of situations *in the story*, not the order *in the text*. Therefore, a notion of story time clearly exists in the model.

Whether or not a proposition p is believed to be the case at story time step t, is indicated by a value $x_{p,t}$. If p at t, denoted p_t, is believed to occur, its value $x_{p,t}$ is equal to 1. If p_t is not believed to be the case, $x_{p,t} = 0$. Since all propositions that constitute story situations are taken from a set of n different propositions, the situation at t can be represented by the vector $X_t = (x_{p,t}, x_{q,t}, \ldots)$ containing n binary elements. This corresponds to a point in an n-dimensional space called the *situation state space* (Golden & Rumelhart, 1993, p. 296). Since each dimension of this space (or, equivalently, each element of the vector) represents one proposition, this is a localist representation.

A story is a temporal sequence of story situations, which is represented as a sequence of points in situation state space. A story describing T situations is represented by the tuple of story situation vectors $\tilde{X} = \langle X_1, X_2, \ldots, X_T \rangle$ called the *story trajectory* through situation state space. Initially, this trajectory does not include any inferences, which means that $x_{p,t}$ only equals 1 if the story text states p_t explicitly.

The objective of the Golden and Rumelhart model is to take such a story trajectory and adjust it to include propositions that were not mentioned in the text, but are likely to be the case considering the statements that were given. An example might clarify this. Consider a story world consisting of just four propositions (so $n = 4$): BOB IS HUNGRY (p), BOB EATS MUCH (q), BOB HAS STOMACH ACHE (r), and BOB IS SATISFIED (s).[1] A typical story over $T = 3$ time steps in this story world might read: *Bob is hungry. He eats a lot, and gets a stomach ache.* This story's trajectory, shown in Table 3.1, consists of one situation vector for each time step ($t = 1, 2, 3$). Each of these vectors has one element for each of the four propositions (p, q, r, s), so the situation space is 4-dimensional.

Table 3.1: Trajectory over $T = 3$ time steps of an example story in a story world consisting of four propositions. The situation vector at time step t is $X_t = (x_{p,t}, x_{q,t}, x_{r,t}, x_{s,t})$.

t	$x_{p,t}$	$x_{q,t}$	$x_{r,t}$	$x_{s,t}$	proposition
1	1	0	0	0	BOB IS HUNGRY
2	0	1	0	0	BOB EATS MUCH
3	0	0	1	0	BOB HAS STOMACH ACHE

Using the knowledge that eating causes satisfaction, a reader of this story might infer that Bob not only has a stomach ache at $t = 3$, but is also satisfied. Ideally, the trajectory corresponding to such an interpretation of the story will look like the one in Table 3.2.

Table 3.2: Trajectory of an interpretation of the example story of Table 3.1, in which it is inferred that Bob is satisfied at $t = 3$. The situation vector at time step t is $X_t = (x_{p,t}, x_{q,t}, x_{r,t}, x_{s,t})$.

t	$x_{p,t}$	$x_{q,t}$	$x_{r,t}$	$x_{s,t}$	inference
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	**1**	BOB IS SATISFIED

[1] The propositions are not given in the usual PREDICATE(ARGUMENT) format because the Golden and Rumelhart model does not make use of a proposition's predicate-argument structure. It is only concerned with propositions as statements carrying truth values.

3.1.2 World knowledge assumptions

Since causal constraints hold among situations, some sequences of situations (or, equivalently, story trajectories) are more likely to occur than others. According to Golden and Rumelhart (1993, p. 206), knowledge about the world is essentially knowledge of the relative probabilities of trajectories. For example, knowing that eating a lot can cause a stomach ache, but that having a stomach ache causes not eating much at all, comes down to knowing that the situation sequence $\langle(\text{BOB EATS MUCH})_{t-1}, (\text{BOB HAS STOMACH ACHE})_t\rangle$ has a much larger probability than does $\langle(\text{BOB HAS STOMACH ACHE})_{t-1}, (\text{BOB EATS MUCH})_t\rangle$. If a reader who has this knowledge comes to believe that Bob ate a lot, the reader will believe Bob's stomach to hurt at a later moment in story time. It is this kind of knowledge about the influence propositions have on one another's probabilities, on which the knowledge of probabilities of situation sequences (i.e., trajectories) is based.

Although this is not stated explicitly, the implementation of world knowledge in the Golden and Rumelhart model is based on the following four simplifying assumptions, visualized in Figure 3.1:

Single propositions What is modeled is how belief in a single proposition influences belief in another single proposition. The influence between beliefs in situations (i.e., combinations of propositions) only emerges as the result of these influences between pairs of propositions from the situations.

Consistency over time Causal knowledge does not depend on the moment in the story. Although the reader's belief in the occurrence of propositions fluctuates over story time, the way propositions influence each other are 'laws of nature' that remain constant.

Range of influence Beliefs in propositions at story time step t are influenced only by beliefs regarding the neighboring time steps $t-1$ and $t+1$. Propositions at other time steps can only have an indirect influence if they leave an effect on the propositions at $t-1$ or $t+1$.

Symmetry Causal knowledge is not directed in time: The influence belief in p_{t-1} has on belief in q_t is the same as the influence in the opposite direction (of q_t on p_{t-1}). This might seem odd at first, since causality clearly is directed in time. It must be kept in mind, however, that p_{t-1} having a positive influence on q_t does not mean that p causes q, but that p_{t-1} and q_t cause *belief* in each other. A stomach ache does not cause having eaten too much, but observing a stomach ache does cause us to believe that too much was eaten before.

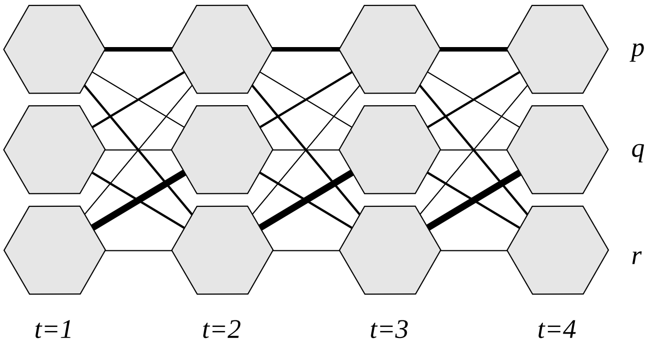

Figure 3.1: Architecture of the Golden and Rumelhart model, in a story world consisting of $n = 3$ propositions (p, q, r) and $T = 4$ time steps $(t = 1, \ldots, 4)$. Every row corresponds to one proposition and every column corresponds to one time step. Links denote influences between propositions-at-time-steps, and their thickness shows the magnitude of this influence. The four world-knowledge assumptions are visualized as follows: There are links between individual propositions only (single propositions); They are the same for all time steps (consistency over time); Propositions at t are linked only to those at $t-1$ and $t+1$ (range of influence); There is no difference between a link from p_{t-1} to q_t and the reverse link from q_t to p_{t-1} (symmetry).

3.1.3 Markov random fields

The mathematics of the Golden and Rumelhart model is based on Markov random field (MRF) theory. A simplified introduction to this theory, applied to the Golden and Rumelhart model, is given here. For a more thorough explanation, see for instance Golden (1996, chap. 6.3) or Cressie (1991, chap. 6.4).

Suppose we have m random variables $z_1, \ldots, z_m$, all real valued in the interval $[0, 1]$. If the probability of the value of z_i is dependent on the value of z_j, then z_i is said to be connected to z_j. Note that, if z_i is connected to z_j, then z_j is also connected to z_i. Since a value depends on itself, all variables are connected to themselves. Such a system is called a Markov random field if every configuration of values $Z = (z_1, \ldots, z_m)$ has a positive probability density. Since probability densities can be arbitrarily close to 0, this is not a serious restriction.

Any particular configuration of values Z is an instantiation of a Markov random field and has associated with it a probability density $P(Z)$. Since this refers to a complete instantiation, it is a *global* probability density. It is not easy to compute, but we can compare the probability densities of two instantiations, as is explained next.

The Hammersley-Clifford theorem (1971, unpublished) as described in Besag (1974) states how a valid probability distribution over a Markov random field can be constructed. First, we need to define the notion of a clique: *A* clique *is a set of variables that are all connected to each other.* Since variables are connected to themselves, every single variable forms a clique. Now let Z_1 and Z_2 be two instantiations of a Markov random field. The Hammersley-Clifford theorem states that $P(Z)$ forms a valid probability density function if and only if

$$\frac{P(Z_1)}{P(Z_2)} = \mathrm{e}^{Q(Z_1)-Q(Z_2)} \tag{3.1}$$

where, for $Z = (z_1, \ldots, z_m)$, function Q has the form:

$$\begin{aligned} Q(Z) = & \sum_{i=1}^{m} z_i G_i(z_i) \\ & + \sum_{i=1}^{m} \sum_{j>i}^{m} z_i z_j G_{ij}(z_i, z_j) \\ & + \sum_{i=1}^{m} \sum_{j>i}^{m} \sum_{k>j}^{m} z_i z_j z_k G_{ijk}(z_i, z_j, z_k) \\ & + \\ & \vdots \\ & + z_1 z_2 \ldots z_m G_{1,2,\ldots,m}(z_1, z_2, \ldots, z_m). \end{aligned} \tag{3.2}$$

Here, the G's are functions such that $G_{ij\ldots}(z_i, z_j, \ldots) = 0$ if variables $z_i, z_j, \ldots$ do not form a clique. So, for all cliques consisting of variables $z_i, z_j, \ldots$ (with $i < j < \ldots$ to make sure that every clique is counted only once) we take the products of $z_i, z_j, \ldots$ and $G_{ij\ldots}(z_i, z_j, \ldots)$. The sum of all these products equals $Q(z_1, \ldots, z_m)$.

For use in the Golden and Rumelhart model, the variables z in Equation 3.2 correspond to propositions-at-time-steps, which are the nodes of the network in Figure 3.1. Consequently, the equation can be greatly simplified. First, it can easily be seen from Figure 3.1 that there exist no fully connected groups (cliques) of more than two nodes. This means that every G-function with more than two arguments equals 0 and these lines disappear from Equation 3.2. Only the first two lines are left over.

Second, by the 'consistency over time' assumption concerning world knowledge, the strength of dependencies between variables is the same for all story time steps. This means that not all connected pairs of variables need to be stated in Equation 3.2 separately because they can be summed over all time steps. The variables z_i and z_j are therefore replaced by variables $x_{p,t-1}$ and $x_{q,t}$ that have additional time step indices. Instead of summing over m variables, we can now sum over n propositions and T time steps.

Finally, it is assumed that a variable's contribution to Q increases linearly with its value. This is accomplished by turning the G-functions into constants. The first line of Equation 3.2 gives rise to n of these: $G_1, \ldots, G_n$, which will be denoted by the vector $B = (b_1, \ldots, b_n)$. The second line of Equation 3.2 gives rise to $n \times n$ constants $G_{11}, \ldots, G_{nn}$, which form a matrix that will be denoted $W = (w_{pq})_{p,q=1,\ldots,n}$. The resulting, simplified Q function is

$$\begin{aligned} Q(\bar{X}) &= \sum_{t=1}^{T} \sum_{p} x_{p,t} b_p + \sum_{t=1}^{T} \sum_{p} \sum_{q} x_{p,t-1} x_{q,t} w_{pq} \\ &= \sum_{t=1}^{T} (B X_t' + X_{t-1} W X_t'). \end{aligned} \tag{3.3}$$

The xs refer to a trajectory $\bar{X} = \langle X_1, X_2, \ldots, X_T \rangle$ consisting of T time steps. The column vector X_t' is the transpose of the row vector X_t. For the equation to be valid at $t = 1$, all values at the non-existent time step $t = 0$ are defined to be 0.

3.1.4 World knowledge implementation

Taking Golden and Rumelhart's assumptions about world knowledge into account, it follows from Markov random field theory that any vector B and matrix W define a probability distribution over all propositions at any number of time steps. The particular values of B and W specify one probability distribution and thereby implement world knowledge. The a priori belief that a proposition p is the case follows from the value b_p, which is therefore called the *bias* of p. The bias vector $B = (b_p, b_q, \ldots)$ contains the biases of all n propositions. The value w_{pq} implements the influence that beliefs in p_{t-1} and q_t have on each other. A positive value of w_{pq} indicates that belief in either p_{t-1} or q_t increases belief in the other. A negative value of w_{pq} indicates the opposite: Belief in p_{t-1} or q_t decreases belief in the other. If w_{pq} equals 0, no causal relation between p_{t-1} and q_t is known to exist. The $n \times n$-matrix W that contains all these values is called the *causal world knowledge matrix*, or world knowledge matrix for short.

The actual values in B and W mainly result from the modeler's intuition. When the model is to process a story, the modeler first of all decides which propositions may be inferred during its comprehension. These propositions, together with those stated in the story text, form the n dimensions of the situation state space. Next, the modeler determines between which of the n propositions there exist causal relations (Golden & Rumelhart, 1993, p. 211; Golden et al., 1994, p. 294). Following this, the values in B and W are either directly set to reflect these relations (Golden & Rumelhart, 1993, pp. 213-214) or they are partly determined on the basis of one typical sequence of situations, called the training trajectory (Golden et al., 1994, pp. 294-296). In the latter case, the values of B are adjusted during a training process until they reflect the probabilities that propositions occur in the training trajectory, while the values of W are to reflect both the assumed causal relations between propositions and the regularities in the succession of training-trajectory situations.

3.1.5 Local probability

If the story text does not mention proposition p at story time step t, this does not mean that p_t is not the case. With enough supporting evidence, the reader may come to believe that p_t is actually quite likely, and infer it. Similarly, the Golden and Rumelhart model infers propositions-at-time-steps that are probably the case, taking into account the rest of the story trajectory. Whether or not p_t is inferred depends on its *local probability*, which is the probability that proposition p occurs at time step t, given the rest of the story trajectory.

Let $\bar{X}^0_{p,t}$ denote the trajectory of a story in which p_t is not the case (i.e., $x_{p,t} = 0$) and $\bar{X}^1_{p,t}$ the trajectory of the same story except that p_t does occur (i.e., $x_{p,t} = 1$). The ratio of the *global* probability densities $P(\bar{X}^0_{p,t})$ and $P(\bar{X}^1_{p,t})$ can be computed from Equations 3.1 and 3.2. In Appendix B.2 it is shown that this ratio equals the ratio of the corresponding *local* probability densities of $x_{p,t} = 0$

and $x_{p,t} = 1$, given the rest of the trajectory. From this, it follows that the local probability of p_t equals

$$\Pr(p_t) = \left(1 + \mathrm{e}^{-\Delta Q_{p,t}}\right)^{-1} \tag{3.4}$$

with

$$\begin{aligned} \Delta Q_{p,t} &= Q(\bar{X}^1_{p,t}) - Q(\bar{X}^0_{p,t}) \\ &= b_p + X_{t-1}W_{\cdot p} + W_{p\cdot}X'_{t+1}. \end{aligned} \tag{3.5}$$

Here, $W_{\cdot p}$ is the column of the world knowledge matrix that contains knowledge about the causes of p, and $W_{p\cdot}$ is the row of W containing knowledge about p's consequences. In case $t = 1$ or $t = T$ (T being the number of situations in the trajectory $\bar{X}$), it is defined that X_0 and X_{T+1} consist of 0s only.

To summarize, Equations 3.4 and 3.5 show how to compute the probability that a proposition p is the case at time step t of a story, given the propositions that are the case at $t-1$ and $t+1$, and world knowledge encoded in vector B and matrix W. The local probabilities are used to infer what is likely to have occurred in the story, given the events that were explicitly stated in the story text and those inferred before.

3.2 Model processing

3.2.1 Inference

According to the Golden and Rumelhart model, comprehending a story comes down to finding the most likely story trajectory given the constraints put by the story. These constraints are formed by the story's statements: If p_t is stated, the value of $x_{p,t}$ is set to 1 and is not allowed to change because it stands for a given fact that cannot be denied. If p_t is not stated, $x_{p,t} = 0$ initially and the probability of p_t, given the rest of the trajectory, needs to be estimated. The initial values, which correspond to the information given in the story text, form the initial trajectory $\bar{X}(0)$. It contains all story time steps, so inference is not modeled as an incremental process in which successive story statements are integrated into the mental representation one by one. The situations do have a temporal ordering, however, given by their story time step indices t.

The probability that proposition p occurs at story time step t, given the rest of the trajectory, is the local probability $\Pr(p_t)$ of Equation 3.4. If it is larger than .5, it is more likely that p_t is the case than that it is not, and the value of $x_{p,t}$ should be set to 1, indicating that p_t is inferred. Conversely, if $\Pr(p_t)$ is less than .5, the value of $x_{p,t}$ should remain (or become) 0, unless p_t was given by the story text. Of course, the local probabilities of *all* propositions at *all* time steps need to be estimated (except for the ones stated in the text). However, changing a single $x_{p,t}$ will generally change the local probabilities of other propositions. To account for this, the values $x_{p,t}$ are not directly set to 0 or 1, but are iteratively adjusted over a number of processing cycles.

The trajectory after $c + 1$ cycles of the inference algorithm is computed from the one in the previous cycle according to

$$x_{p,t}(c+1) = \left[x_{p,t}(c) + \beta \Delta Q_{p,t}\right]_0^1 \tag{3.6}$$

for all p_t not stated in the story. A parameter β, set to a value of $\beta = 0.1$, controls the rate of change in $\bar{X}$. The square brackets with indices 0 and 1 indicate that, whenever the enclosed expression is negative or larger than 1, it is set to 0 or 1, respectively.

It follows from Equation 3.4 that $\Delta Q_{p,t}$ is positive whenever $\Pr(p_t) > .5$ and negative when $\Pr(p_t) < .5$. As a result, the value of $x_{p,t}$ increases to a maximum of 1 if p_t is more likely the case than not, and it decreases to a minimum of 0 if p_t if more likely not to occur. All the values in the trajectory are repeatedly updated in parallel according to Equation 3.6 until they no longer change (Golden et al., 1994, p. 297). Golden (1993) proves that this process does indeed converge, provided β is chosen small enough. The resulting trajectory forms the model's interpretation of the story.

3.2.2 Recall

The model is also used to simulate free recall of stories. This process is modeled in two phases. First, an attempt is made to reconstruct the story trajectory. Next, the resulting, reconstructed trajectory is used to determine which propositions are recalled by the model.

The reconstruction process is similar to the inference process described above. The main difference is that the result of trajectory reconstruction depends not only on the world knowledge implemented in B and W, but also on the trajectory that is to be recalled, which we shall denote $\bar{X}^*$. This trajectory can be viewed as an episodic memory trace that becomes weaker as the amount of time between reading and recall of the story (the so-called retention interval) increases. In the model, retention interval is controlled by a strictly positive parameter ρ. For short retention intervals ($\rho \approx 0.1$), the memory trace is strong and dominates the reconstruction process. As a result, the reconstructed trajectory will be quite similar to $\bar{X}^*$. As ρ gets larger, the memory trace weakens and the reconstructed trajectory depends more on world knowledge and less on the trajectory to be recalled.

Trajectory $\bar{X}^*$ serves as "a set of 'soft' constraints" (Golden et al., 1994, p. 296) that guide the reconstruction process to result in a trajectory that is like $\bar{X}^*$. This is implemented by adding to the value of $\Delta Q_{p,t}$ (of Equation 3.5) a term that depends on the difference between the current trajectory $\bar{X}$ and $\bar{X}^*$:

$$\Delta Q_{p,t} = b_p + X_{t-1}W_{\cdot p} + W_{p\cdot}X'_{t+1} + \tfrac{1}{\rho}\left(x^*_{p,t} - x_{p,t}\right). \tag{3.7}$$

Reconstruction progresses by repeatedly applying Equation 3.6, with $\Delta Q_{p,t}$ defined as in Equation 3.7, until the trajectory no longer changes.[2]

The model's free recall of the story depends on the trajectory resulting from

[2] In Golden et al. (1994), the last term of Equation 3.7 is claimed to be $\frac{1}{\rho}(x_{p,t} - x^*_{p,t})$, which cannot be correct since it would cause the recalled trajectory to be as different as possible from the to-be-recalled trajectory.

the reconstruction process. For each story time step t, the proposition with the largest value of $x_{p,t}$ is assumed to summarize the situation at t. If none of the values at a certain t exceeds .5, no proposition is recalled for that time step (Golden & Rumelhart, 1993, p. 215).

One of the important details that remain unclear is whether $\bar{X}^*$ equals the original story trajectory $\bar{X}(0)$ or the final result of the inference process applied to $\bar{X}(0)$. According to Golden et al. (1994) "First, a story is 'comprehended' by the model by computing the most probable trajectory .... Then the story is 'recalled' by the model using the model's 'interpretation' of the text ... as a set of 'soft' constraints" (p. 296). Although this clearly states that trajectory $\bar{X}^*$ is the result of the model's inference process, one page later it is claimed that the to-be-recalled trajectory is the original story trajectory: "The initial trajectory [$\bar{X}(0)$] is a partial specification of a text .... To use the algorithm to model the recall process, ... the constraint matrix is set equal to the initial trajectory" (p. 297). This issue is not cleared up by Golden and Rumelhart (1993). They first state that "story recall is viewed as another process that attempts to retrieve the *constructed* [italics added] trajectory from memory" (p. 208), while at a later point it is claimed that during the reconstruction process "the constraints imposed by the retention-interval parameter help because they restrict the search to trajectories close ... to the *original* [italics added] story trajectory" (p. 215).

Also, it is not made clear which trajectory provides the initial values for the reconstruction process. Although it is stated that the value at $t = 1$, $t = 2$, and $t = 3$ are not allowed to change during the reconstruction process because they form the retrieval cue for the rest of the trajectory (Golden & Rumelhart, 1993, p. 215), it is not mentioned what these or any other values are initially.

3.3 Evaluation

3.3.1 Inference

Unlike the Gestalt models, the Golden and Rumelhart model has a notion of processing time because a variable number of processing cycles are needed to find the most likely trajectory. However, this number cannot be taken as a measure of reading time. Reading is a process in which clauses from a text are processed one by one. The time it takes to read one clause depends on many variables, among which the causal relation between the clause and the story read so far (Myers, Shinjo, & Duffy, 1987; Sanders & Noordman, 2000). The Golden and Rumelhart model, however, starts comprehension of a story with the *complete* story trajectory. It takes an amount of processing time to come up with an interpretation of the story, but this processing time cannot be related to a reading time of any of the individual clauses of the story. In fact, the model is not validated against any empirical data concerning inference at all.

This does not mean that the model is necessarily inappropriate for simulating inferences during story comprehension. However, only in Golden et al. (1994) are results presented, and these are based on just one text: the 'Epaminondas' story previously used by Trabasso, Secco, and Van den Broek (1984) to examine the effect of causal relatedness on recall of story statements. The first six sentences of this story read:

> Once there was a little boy who lived in a hot country. One day his mother told him to take some cake to his grandmother. She warned him to hold it carefully so it wouldn't break into crumbs. The little boy put the cake in a leaf under his arm and carried it to his grandmother's. When he got there, the cake had crumbled into tiny pieces. His grandmother told him he was a silly boy and that he should have carried the cake on top of his head so it wouldn't break. (Golden et al., 1994, p. 300)

Golden et al. (1994, Table 11.5) parse the Epaminondas story into 20 propositions. Nine propositions are added to these because they need to be inferred for adequate comprehension of the story, so the situation state space consists of $n = 29$ dimensions. For instance, from MOTHER TELLS BOY TO BRING CAKE TO GRANDMOTHER it can be inferred that BOY WANTS TO BRING CAKE TO GRANDMOTHER. Table 3.3 shows the 11 story propositions taken from the first six

Table 3.3: Fifteen propositions and the story time steps in which they first occur (or are inferred to occur), corresponding to the first six sentences of the Epaminondas story. Four of the propositions are labelled ⋆ to indicate that these are not part of the text but are added for the model to infer (adapted from Golden et al., 1994, Table 11.5).

t	nr.	proposition
1	1	BOY EXISTS
	2	BOY IS LITTLE
	3	BOY LIVES IN HOT COUNTRY
2	4	MOTHER TELLS BOY TO BRING CAKE TO GRANDMOTHER
	⋆ 5	BOY WANTS TO BRING CAKE TO GRANDMOTHER
3	6	MOTHER TELLS BOY TO HOLD CAKE CAREFULLY
	⋆ 7	BOY WANTS TO HOLD CAKE CAREFULLY
4	8	BOY PUTS CAKE IN LEAF UNDER ARM
5	9	BOY CARRIES CAKE TO GRANDMOTHER
	⋆ 10	CAKE BREAKS INTO CRUMBS
6	11	BOY ARRIVES AT GRANDMOTHER'S
	⋆ 12	BOY IS AT GRANDMOTHER'S
7	13	CAKE IS IN CRUMBS
8	14	GRANDMOTHER TELLS BOY THAT BOY IS SILLY
9	15	GRANDMOTHER TELLS BOY TO PUT CAKE ON BOY'S HEAD

sentences of the Epaminondas story, and the 4 inferable propositions that are added to these.

All of the 29 propositions are used for the construction of a training trajectory consisting of 14 time steps. Together with assumptions about causal relations between the propositions, this training trajectory is used for finding values of bias vector B and world knowledge matrix W.

Next, the inference model processes a test story that consists of a sequence of 20 time steps using 20 different propositions from the 29 in the situation state space. For unknown reasons, only 17 of these 20 propositions are part of the original Epaminondas story while the other 3 come from the 9 propositions that were added to be inferred. The model is able to correctly fill in information that is implied by the story. For instance, the boy is said to arrive at his grandmother's house at $t = 6$ (proposition number 11 in Table 3.3). At $t = 14$, he goes back to his mother's. In the meantime ($t = 7, \ldots, 13$), he is inferred to be at his grandmother's (proposition 12), even though the story does not explicitly state this (Golden et al., 1994, Figure 11.8).

However, the model also infers propositions that do not follow from the story at all. The reason for this is that its world knowledge is based on a single training trajectory that is very specific and cannot be said to contain any knowledge apart from what is directly needed for understanding the Epaminondas story. As a result, *B* and *W* are so specifically tailored to this story that it is almost impossible to infer anything else: Any story is interpreted as being almost equal to the Epaminondas training trajectory. For instance, the test story states that there once was a boy without including the information that he is little and lives in a hot country. However, the model infers these two additional facts because the only boy it knows anything about was small and lived in a hot country. Had the world knowledge in *B* and *W* also been based on training trajectories about larger boys in colder places, these superfluous inferences might not have been made.

The model also processes a second, shorter test story, consisting of only 12 propositions. None of these states that the boy's grandmother told him to carry the cake on his head. Nevertheless, this is inferred by the model because, according to its world knowledge, this is what grandmothers always say (Golden et al., 1994, Figure 11.9 and Table 11.5).

3.3.2 Recall

For simulating free recall, Golden and Rumelhart (1993) use the Epaminondas story described above and three more stories ('Judy's birthday', 'Tiger's whisker', and 'Fox and bear') all taken from Trabasso et al. (1984). The model's results correspond to Trabasso et al.'s in several respects. First, and least surprisingly, the number of recalled story propositions decreases as retention interval (parameter ρ) increases. Second, if the four stories are ordered by percentage of story propositions recalled, the model and the Trabasso et al. data result in the same order. Third, story propositions with more causal connections to other story propositions are recalled more often than propositions with lower causal connectivity. The magnitude of this effect increases as ρ increases (Golden & Rumelhart, 1993, pp. 218-223). A fourth result is concerned with the order of events in the recalled story. Bischofshausen (1985) had subjects recall stories in which the causally sensible order of story events was different from the order in which the events were presented in the story text. She found that the recalled stories often followed the causal order instead of the textual order.

Moreover, this effect increased with retention time. To simulate this, Golden and Rumelhart had the model recall distorted versions of the four stories used before, in which every story had two of its time step indices swapped. In the recalled trajectories, these misplaced situations were sometimes found at the original time steps instead of at their stated positions. This occurred more often as retention interval parameter ρ was increased (Golden & Rumelhart, 1993, p. 223).

3.4 Conclusion

Even though the Golden and Rumelhart model does not predict processing times for individual story time steps, it does away with the two shortcomings of the Story Gestalt model by representing story time explicitly and by implementing inference as an iterative process that requires a certain amount of processing time. However, the model introduces a new problem: Its architecture seriously limits the stories and world knowledge that can be represented. Three main limitations are:

Constraints within a time step The 'range of influence' assumption concerning the implementation of world knowledge (see Section 3.1.2) states that values at time step t are influenced only by those at $t-1$ and $t+1$. However, it may be necessary to impose constraints on propositions *within* a time step. For instance, a story character might have reached a road junction at time step $t-1$, which will cause her to make a left or right turn at t. She cannot turn both left and right, which is a constraint within t.

Disjunction A story statement that is a conjunction of two propositions, like *it is raining and cold*, is represented by setting both $x_{\text{RAIN},t} = 1$ and $x_{\text{COLD},t} = 1$. For disjunctions, however, this is not possible. A statement like *the butler or the mysterious stranger committed the crime* can only be represented as a single proposition, in which case the *or* is no longer an operator that combines two propositions.[3]

Combined effect of propositions Minsky and Papert (1969) show that an architecture like Golden and Rumelhart's cannot compute the exclusive or (XOR) function. Suppose both p and q are stated at time step $t-1$, so $x_{p,t-1} = x_{q,t-1} = 1$. The influence that the conjunction of p_{t-1} and q_{t-1} has on a proposition r at time step t equals the sum of the influences of p_{t-1} and q_{t-1} individually. This follows from Equation 3.5: Assuming that all other values equal 0, $\Delta Q_{r,t} = b_r + w_{pr}x_{p,t-1} + w_{qr}x_{q,t-1} = b_r + w_{pr} + w_{qr}$. Therefore, it is impossible to represent the knowledge that, for instance, taking *either* medicine A or B can cure a disease but taking *both* at the same time will

[3] In the field of propositional logic, the De Morgan law states how disjunction can be rewritten in terms of negation and conjunction: $p \vee q \equiv \neg(\neg p \wedge \neg q)$. However, this does not help here, since the conjunction $\neg p \wedge \neg q$ must be represented as a single proposition in order to be negated.

make things worse, since this would require that w_{pr} and w_{qr} are positive while their sum is negative.

These three limitations result from the 'single propositions' and 'range of influence' assumptions regarding world knowledge, and from the representation of story situations. In this localist representation, each of a number of predetermined propositions corresponds to one dimension of the situation state space. As a result, a situation can only be a conjunction of propositions. There is a dimension for proposition p and one for q, but if the disjunction $p \vee q$ needs to be represented, it requires its own dimension. In that case, however, since there are no constraints among propositions within a time step, the value of $x_{p \vee q,t}$, indicating whether the disjunction is the case at t, is independent of the values $x_{p,t}$ and $x_{q,t}$. It is therefore possible that p_t is known to be the case, but its logical implication $(p \vee q)_t$ is not.

This shows that it is the model's absence of knowledge about relations within a time step that is, at least partly, responsible for its inability to represent disjunctions. This inability for its part implies that an XOR cannot be represented either, since the XOR operator can be rewritten as a disjunction: $p \text{ XOR } q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$. It is necessary to represent $p \text{ XOR } q$ in a single situation-space dimension for the implementation of knowledge about causal relations requiring the XOR function, because the 'single proposition' assumption excludes causal knowledge that is based on the belief in some combination of the individual propositions p and q.

Considering this, it may seem as if all three representational problems can be solved by simply adding world knowledge about the relations between propositions within a time step, retracting the 'range of influence' assumption. As a result, the model's architecture will look something like the network in Figure 3.2, which is similar to Figure 3.1 except that links are added between each pair of propositions within the same time step.

Although this may seem only a minor adjustment, the consequences for the MRF analysis are considerable. It is easy to see that adding links within a time step results in the occurrence of more and larger cliques. Apart from cliques of two propositions at neighboring time steps, which existed before, there are now cliques of two propositions at the same time step, cliques of three propositions at one time step or at two neighboring time steps, and cliques of four, five, and six propositions at two neighboring time steps. In general, in a story world consisting of n propositions, there are cliques of up to $2n$ nodes, which consist

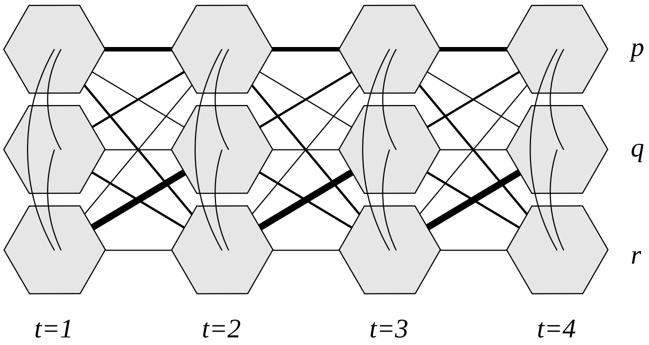

Figure 3.2: Architecture of the Golden and Rumelhart model after adding relations between propositions within a time step.

of all propositions at two adjacent time steps. For the Q-function (Equation 3.2) this means that it can hardly be simplified anymore and the number of parameters G that need to be estimated becomes unpractically large. In short, simply adding influences between propositions within a time step does not mean a simple addition of a bit more world knowledge in the form of one $n \times n$-matrix. On top of this, there need to be three 3-dimensional $n \times n \times n$-arrays, four 4-dimensional arrays, and many, many more.

Adding all these parameters is especially problematic because of a property that the Golden and Rumelhart model shares with the Construction-Integration model: World knowledge is, at least in part, decided upon by the modeler, introducing a source of subjectivity into the model. Moreover, the world knowledge that is applied in practice is often very specific for the story that is processed instead of being applicable more generally. A better approach would be to construct a microworld that is complex enough for many different stories to take place in, not just for different versions of the same story. With enough training material in the form of trajectories corresponding to stories in the microworld, it would be possible to obtain a bias vector and a causal world knowledge matrix without direct intervention by the modeler.

As we have seen in Section 2.6, such a method is used by the Story Gestalt model to develop distributed representations of propositions. Both the Story Gestalt model and the Golden and Rumelhart model take as input a story in the form of a sequence of propositions and add to this propositions that are inferred also to be the case in the story, using knowledge about the world in which the stories take place. Because of this similarity in modeled task, the Golden and Rumelhart model is more comparable to the Story Gestalt model than any other model discussed in the previous chapter.[4] Interestingly, the two models are quite different in architecture. Their most noticeable differences are listed in Table 3.4.

Table 3.4: Comparison between the Story Gestalt model and the Golden and Rumelhart model.

	Story Gestalt	Golden and Rumelhart
Representation	distributed	localist
World knowledge	microworld	hand-coded
Processing time	no	yes
Story time	no	yes

Applying the Story Gestalt model's microworld method to the Golden and Rumelhart model also opens up the possibility of using the microworld training set to develop distributed representations, which can encode world knowledge additional to vector B and matrix W. In this way, it might be possible to implement knowledge about constraints within a time step while keeping the simple MRF analysis. In the following chapter, we shall present the Distributed Situation Space model that does exactly this, thereby combining the best of both models. It has notions of processing and of story time, but it uses distributed representations that are based on knowledge of a microworld. Moreover, unlike either the Story Gestalt or the Golden and Rumelhart model, it not only accomplishes the inference task but also predicts empirical data concerning inference.

[4] The Construction-Integration model also takes a set of propositions as input and adds inference, but its need for a modeler to intervene subjectively during this process disqualifies it as a computational model.

4 The Distributed Situation Space model

An article based on this chapter, Section 5.1, and parts of Chapter 3, appeared as Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003b), Modeling knowledge-based inferences in story comprehension, *Cognitive Science, 27*, 875-910.

In Section 1.1.2, it was discussed that Kintsch and Van Dijk (1978) assume three levels of text representation: the surface text level, the textbase level, and the situational level. The first consists of the text's literal wording, while the textbase is often regarded as a network of propositions from the text. When these are integrated with the reader's knowledge, a situational representation results. The Distributed Situation Space (DSS) model, presented in this chapter, aims at representing stories at this situational level, while ignoring any textbase-like representation. Therefore, the model's focus is at *knowledge* instead of *text*.

When comprehension of a text requires concepts or propositions that originate from the reader's knowledge rather than from the text being processed, these can be added to the text representation of the Resonance, Landscape, and Construction-Integration models. However, this is done on an ad hoc basis by the modeler and not by a process within the model itself. The same is true of causal knowledge in the Langston and Trabasso model. As we have seen, the Predication model is not able to create knowledge about propositions from the knowledge about word meaning present in LSA. Of the seven models discussed in Chapter 2, only the Story Gestalt model simulates how a text selects relevant knowledge and how inferences come about, as does the Golden and Rumelhart model of the previous chapter. It therefore comes as no surprise that it is these two models to which the DSS model is most similar.

With the Story Gestalt model, the DSS model has in common that the amount of knowledge to be implemented is made manageable by letting stories take place in a microworld. The setup of this microworld is described in Section 4.1. Another similarity between the Story Gestalt and DSS model is that situations are represented distributively. Section 4.2 explains how a description of the microworld forms the basis of such a representation.

With the Golden and Rumelhart model, the DSS model shares most architectural assumptions and its mathematical basis, from which it follows how world knowledge concerning relations between story time steps is implemented (discussed in Section 4.3) and how this knowledge is applied to the story representation in order to result in inferences (Section 4.4). Also, the idea of a 'situation space' in which story situations are represented is an important aspect of both models. It is from this space and its distributed nature that the DSS model gets its name.

In Section 4.5, the model processes three stories to show it can deal with different types of inference problems. Also, aggregated results on many stories are compared to experimental findings. Following this, the last section of this chapter evaluates the model in light of Jacobs and Grainger's (1994) model evaluation criteria.

4.1 The microworld

Any story that the model is to process takes place in a microworld, all knowledge of which is implemented in the model. Although this microworld allows only for very simple stories, it is complex enough to evaluate the model's properties.

For the construction of the microworld, we begin by choosing a small number of basic propositions from which every story can be built up. In our microworld there exist two story characters, who are named Bob and Jilly. Their possible activities and states can be described using the 14 basic propositions shown in Table 4.1.

Table 4.1: Fourteen basic microworld propositions and their intended meanings.

nr.	proposition	meaning
1	SUN	The sun shines.
2	RAIN	It rains.
3	B OUTSIDE	Bob is outside.
4	J OUTSIDE	Jilly is outside.
5	SOCCER	Bob and Jilly play soccer.
6	HIDE-AND-SEEK	Bob and Jilly play hide-and-seek.
7	B COMPUTER	Bob plays a computer game.
8	J COMPUTER	Jilly plays a computer game.
9	B DOG	Bob plays with the dog.
10	J DOG	Jilly plays with the dog.
11	B TIRED	Bob is tired.
12	J TIRED	Jilly is tired.
13	B WINS	Bob wins.
14	J WINS	Jilly wins.

As in the Golden and Rumelhart model, events in the microworld are assumed to follow one another in discrete *story time steps*. At each time step, some propositions occur and others do not. The combination of all propositions that are the case or that are not the case at the same moment in story time is called the *situation* at that time step.

The basic propositions are not unrelated within a time step but put constraints on one another. For instance, two hard constraints are that Bob and Jilly can only play soccer when they are outside and can only play a computer game when inside (which is defined as not-outside). Other important constraints are that Bob and Jilly can only perform one activity at a time and that it is only possible for someone to win when they play soccer, hide-and-seek, or both play a computer game. It goes without saying that no proposition can be the case at the same time as its negation. There also exist soft constraints. For instance, Bob and Jilly are more likely to be at the same place and do the same thing than to be at different places and do different things, and they are more likely to be outside than inside when the sun shines while the reverse is true during rain.

All knowledge about constraints among propositions within a time step is considered non-temporal world knowledge. *Temporal* world knowledge, on the other hand, is concerned with contingencies between (combinations of) propositions at adjacent time steps. Here too, there are hard and soft constraints. Two hard constraints are that Bob and Jilly stop the game they are playing after one of them wins and that a game can only be won if it was played in the previous time step. An important soft constraint is that whoever is tired is less likely to win at the next time step. Also, Bob and Jilly are more likely to stay where they are than to change place unless, of course, the weather changes.

The regularities that hold in our microworld will not be implemented in the model directly. Rather, they are used to construct a realistic sequence of situations from which the world knowledge needed by the model is extracted, as explained in the following section. Observing the temporal relations and the non-temporal constraints, a *microworld description* of 250 consecutive example situations was constructed. In all of these, each basic proposition is stated to be either the case or not the case. For instance, the 14th example situation states that Bob and Jilly are outside playing soccer and do not play another game, that the sun does not shine and it does not rain, and that nobody is tired or wins. The following, 15th example situation is identical except that Bob became tired, which is why Jilly wins in example situation number 16.

4.2 The Situation Space

Every dimension of Golden and Rumelhart's situation space corresponds to exactly one proposition, so propositions are represented locally in this space. The DSS model, on the other hand, uses a distributed representation. As in the Golden and Rumelhart model, a proposition in the DSS model is represented by a vector in a high-dimensional situation space. However, there is no one-to-one correspondence between propositions and dimensions of the distributed situation space. Each dimension codes for several propositions, and each proposition is represented by a combination of values in more than one dimension.

In this section, we will describe a representation in which both the a priori and conditional subjective probabilities of propositions can be directly computed from their vectors. Such a subjective probability is called a *belief value* since it indicates to what extent the proposition is believed to be the case. In this representation, propositions can be combined using the Boolean operators of negation, conjunction and disjunction, while preserving the relation between their representations and belief values.

We will start by presenting a representation in which propositions correspond to areas in a two-dimensional space. From this, the representation for negations, conjunctions and disjunctions of propositions follows naturally and the limitations of the Golden and Rumelhart model mentioned in Section 3.4 are overcome. Next, it will be explained how such a representation can be extracted automatically from the description of microworld events discussed in Section 4.1, and that this representation is equivalent to a representation as points in high-dimensional space. Finally, we will show how belief values can be computed from such a distributed representation.

4.2.1 Representing propositions and situations

Suppose there is a story world consisting of only two propositions: p and q. In the story world, each of these is the case half of the time, or stated as probabilities: $\Pr(p) = \Pr(q) = .5$. If the propositions had been independent of each other, the probability of their conjunction would have been $\Pr(p \wedge q) = \Pr(p)\Pr(q) = .25$. However, in this story world, p and q exclude each other to some extent, causing the probability of their conjunction to be only $\Pr(p \wedge q) = .125$.

Figure 4.1 shows how such a story world can be represented by assigning to p and q particular areas within a rectangular map that confines the space of all possibilities. To each proposition is assigned an area that occupies half of the total map, reflecting the .5 a priori probability of both propositions. For clarity, this is shown for the two propositions separately in the top row of Figure 4.1

A story situation is a (partial) description of events at one moment in the story. The horizontally hatched area corresponds to situations in which p is the case, and the vertically hatched area corresponds to situations that include q. Figure 4.1 also shows how representations can be constructed for more complex situations, which are formed by applying Boolean operators to the two basic propositions p and q. The overlap of the two areas represents situations in which both p and q occur. It occupies $1/8$ of the map, indicating that $\Pr(p \wedge q) = .125$. Note that the more propositions are the case, the smaller the corresponding map area becomes. The area's size indicates the amount of information that is available, with small areas corresponding to much information.

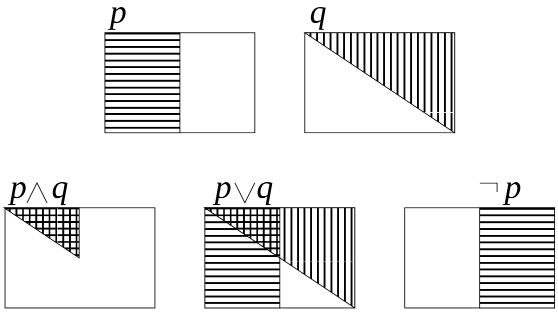

Figure 4.1: Dependencies between two propositions (p and q) represented as two-dimensional areas. Proposition p corresponds to the horizontally hatched area and q corresponds to the vertically hatched area. Propositions can be combined by means of conjunction (bottom left), disjunction (bottom center), or negation (bottom right). The fraction of the map covered by a proposition or combination thereof equals its probability of occurrence in the story world.

Likewise, representations of negations and disjunctions can be constructed while retaining the relation between map area sizes and probabilities. Since a map area corresponds to the occurrence of a proposition, the occurrence of its negation must be exactly the rest of the rectangular map. The fraction of the map covered by the p area equals the probability of p, so the fraction *not* covered by the p area must equal $1 - \Pr(p)$. Indeed, this equals $\Pr(\neg p)$, the probability that p is not the case.

The map area corresponding to the disjunction $p \vee q$ equals the area covered by either one of the propositions. The bottom center map of Figure 4.1 shows that the size of this area equals the sum of the sizes of the p- and q areas, minus the size of the area covered by both (so that this area is not counted twice). This is in accordance with probability theory: The probability of $p \vee q$ equals $\Pr(p \vee q) = \Pr(p) + \Pr(q) - \Pr(p \wedge q)$.

Note that this representation does not allow for a distinction between propositions and situations. Whether a map area corresponds to a basic proposition (p or q) or to some combination of these can only be established by comparing it to representations that were defined as basic propositions (p and q in Figure 4.1). This inability to distinguish propositions from situations is in accordance with our claim that DSS represents stories at Van Dijk and Kintsch's (1983) situational level. Such a representation is similar to the result of experiencing the story events (Fletcher, 1994). Unlike their textual descriptions, experiences are not reducible to separate propositions.

Be reminded that the Golden and Rumelhart model suffers from at least three limitations regarding the representation of story situations and world knowledge (see Section 3.4): Constraints within a time step, disjunctions, and the exclusive-or relation cannot be represented by the Golden and Rumelhart model. Because of the DSS model's distributed representation of propositions described above, it does not have these shortcomings. First, constraints between propositions within the same story time step are implemented in their representations. Second, it is now possible to represent not only conjunctions but also disjunctions. Third, since the representation of $p \wedge q$ is not the sum of the representations of p and q separately, knowledge about the causal effects of the conjunction can be qualitatively different from the combined knowledge about the individual propositions.

4.2.2 Self-Organizing Maps

For any realistic number of propositions, it is impossible to construct by hand a map such that the projections of all propositions on this map correspond even approximately to their interdependencies. Fortunately, this can be done automatically by means of a Self-Organizing Map (SOM), also known as Kohonen Map (Kohonen, 1995). Such a map is a grid of cells that can be made to organize itself to map propositions as described above. Figure 4.2 shows the map used here, consisting of $10 \times 15 = 150$ hexagonal cells. A SOM does not need to be two-dimensional, but this is convenient for visualization. Also, the exact size and shape of the map do not have a large effect on the quality of the mappings that will develop.

Between every two cells i and j, a distance $d(i,j)$ is defined which equals the minimum number of steps needed to get from i to j, if every step takes you from a cell to an immediate neighbor. The distance from a cell to itself is 0, the distance between two neighboring cells is 1, and the largest distance on a 10×15 map with hexagonal cells equals 16, as can be seen from Figure 4.2. The *neighborhood* with size N of cell i is defined as the set of cells that lie within a distance of N from i, so j is in the neighborhood of i if and only if $d(i,j) \leq N$. Note that i is always in its own neighborhood.

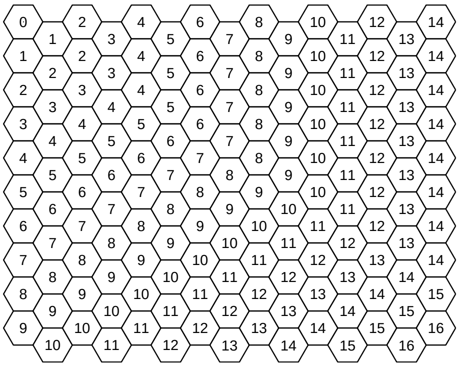

Figure 4.2: A 10×15 Self Organizing Map with hexagonal topology. A cell's number indicates its distance to the top left cell. The neighborhood with size N of the top left cell consists of all cells whose number is less than or equal to N.

Training the Self-Organizing Map

With each SOM-cell i is associated a vector μ_i of values between 0 and 1. These vectors consist of one element for each of the basic propositions from Table 4.1, so $\mu_i = (\mu_i(\textsc{sun}), \mu_i(\textsc{rain}), \ldots, \mu_i(\textsc{j wins}))$. Each value $\mu_i(p)$ is the extent to which cell i is part of the representation of proposition p. Training the SOM comes down to adjusting these membership values such that the non-temporal constraints among microworld propositions are mapped onto the two dimensions of the SOM.[1] If a perfect mapping is not possible, the SOM makes an approximation. The process of obtaining representations of propositions by means of self-organization is not considered part of the psychological model. We do not claim that this is how actual mental representations of propositions develop.

Section 4.1 explained how a microworld description consisting of 250 example situations was constructed. In each of these, every basic proposition is either known to be the case or known to be not the case. Since there are 14 basic propositions, an example situation can be represented by a vector consisting of 14 binary elements, one for each proposition. An element has a value of 1 if the corresponding proposition is the case, or 0 if it is not. For instance, the situation in which the sun shines (the first proposition), Bob is outside (the third proposition), and no other basic proposition is the case, corresponds to the vector $S = (1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

The process of self-organization begins with setting a learning rate parameter $\alpha = .9$, a neighborhood size parameter $N = 16$, and all membership values $\mu_i(p) = .5$. Next, the SOM is trained by repetitively presenting it with all vectors from the microworld description. After presentation of each vector S, the following steps are executed:

1. The euclidean distances between each membership vector μ_i and S are computed. Initially, all these distances are the same because all vectors are identical.
2. Let i be the cell whose vector is closest to S. If there are several cells with the same, minimum distance to S, one of them is chosen at random.
3. All cells in the neighborhood with size N of cell i have their membership vectors moved towards S. If j is one of these cells, its vector μ_j changes to $\mu_j + \alpha(S - \mu_j)$. As a result, the map area that corresponds closest to example situation S is made to represent S even stronger.
4. The neighborhood size N is reduced slightly, decreasing the sizes of the areas influenced by the input vectors.

[1] Temporal contingencies between consecutive situations are ignored in this phase and are dealt with in Section 4.3.

5. The learning rate α is reduced slightly, making the current mapping more stable.

These steps are repeated until all example vectors S have been presented to the SOM 100 times. By then, $N = 6$ and $\alpha = .02$. Next, the training process continues but without changing α. After presenting all example vectors 60 more times, $N = 0$ and training is completed.

The SOM representation of a proposition is obtained by taking from each cell's membership vector the element corresponding to the proposition. For instance, the first value of vector μ_i is $\mu_i(\textsc{sun})$. This is the extent to which cell i belongs to the representation of SUN. Taking the first value of the membership vectors of all cells results in the full representation of 'the sun shines'. Figure 4.3 shows the resulting map for each basic proposition of our microworld.

Representing complex propositions

Since the membership values $\mu_i(p)$ range between 0 and 1, a proposition's area on a SOM is fuzzy instead of sharply defined. Therefore, we need to resort to fuzzy set theory to define the areas corresponding to negations, conjunctions, and other complex propositions. Given a cell's membership values for p and for q, its values for 'not p' and for 'p and q' are computed as follows:[2]

$$\begin{aligned} \mu_i(\neg p) &= 1 - \mu_i(p) \\ \mu_i(p \wedge q) &= \mu_i(p)\mu_i(q). \end{aligned} \tag{4.1}$$

It is a well known fact that all connectives in propositional logic can be defined in terms of negation and conjunction, so any story situation can be represented using the mappings from Figure 4.3 and the rules for combining them in Equation 4.1. For instance, the membership values for the disjunction 'p or q' follow from the De Morgan law:

$$\begin{aligned} \mu_i(p \vee q) &= \mu_i(\neg(\neg p \wedge \neg q)) \\ &= \mu_i(p) + \mu_i(q) - \mu_i(p)\mu_i(q). \end{aligned}$$

[2] This is not the only way to model negation and conjunction in fuzzy logic. In particular, $\mu_i(p \wedge q) = \min\{\mu_i(p), \mu_i(q)\}$ is often used. However, using the product to model conjunction yields Equation 4.3 for computing conditional belief values, which has the useful property that the belief value of a proposition always equals 1 minus the belief value of its negation.

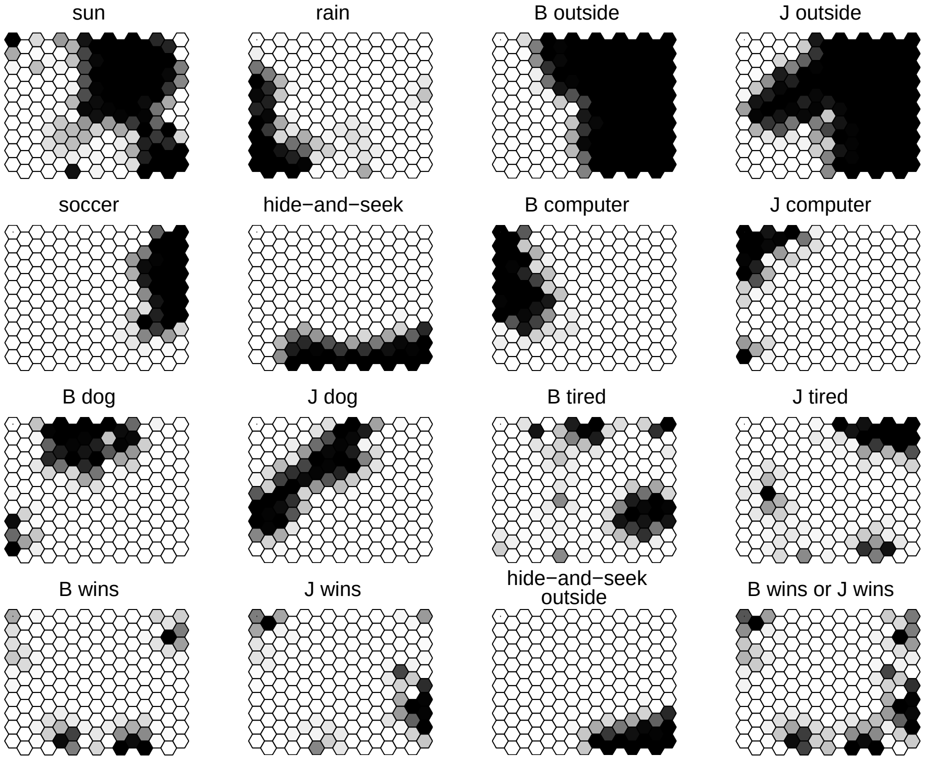

Figure 4.3: Automatically constructed mappings of propositions on a Self-Organizing Map with $n = 10 \times 15$ cells. The darkness of a cell i indicates its membership value $\mu_i(p)$ for the corresponding proposition p. The last two mappings of the bottom row are examples of combined propositions, representing HIDE-AND-SEEK $\wedge$ B OUTSIDE $\wedge$ J OUTSIDE (Bob and Jilly play hide-and-seek outside), and B WINS $\vee$ J WINS (Bob or Jilly wins) respectively.

Likewise, the statement 'either p or q' can be represented by applying the exclusive-or operator: p XOR $q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$. Figure 4.3 shows the mappings of a conjunction and of a disjunction of two basic propositions as an example.

Mappings and vectors

The n cells of the SOM form a two-dimensional grid, but can also be viewed as dimensions of an n-dimensional state space $[0, 1]^n$. Any *area* on the SOM, defined by membership values $\mu_i(p)$ for all cells i, corresponds to a *point* in this distributed situation space, defined by the vector $\mu(p) = (\mu_1(p), \ldots, \mu_n(p))$. It must be kept in mind, however, that the difference between the SOM area and DSS vector representations is purely aesthetic. The vectors are used in mathematical formulas, while the areas are useful for visualization purposes.

From now on, the symbol X shall be used to refer to a vector in situation space, which may or may not equal a proposition vector $\mu(p)$ or a vector representing a complex proposition as computed using Equation 4.1. A story is a temporal sequence of situation vectors, that is, a trajectory through situation space. If $X_t \in [0,1]^n$ is the situation vector at story time step t, the trajectory of a story consisting of T situations is the T-tuple $\bar{X} = \langle X_1, X_2, \ldots, X_T \rangle$. The model takes this trajectory as input and, during the inference process explained in Section 4.4, converts it into a more informative trajectory. How the resulting trajectory can be interpreted is explained next.

4.2.3 Belief values

We now know how to represent any story situation as a vector in situation space. In order to interpret the trajectory that results from the inference process it will also be necessary to take the opposite route: given some vector, reconstruct the situation. This is not generally possible, since only few points in situation space correspond exactly to some combination of propositions. We can, however, compute the belief value of any proposition given a DSS vector.

Let $X = (x_1, x_2, \ldots, x_n)$ be a situation vector (or, equivalently, a SOM area). As an 'abuse of notation', the symbol X will also be used to refer to the situation represented by the vector X. As a result of training the SOM, the subjective unconditional probability that situation X occurs in the microworld equals the fraction of the map that it covers. This value, denoted $\tau(X)$, is the belief value of situation X and equals

$$\tau(X) = \frac{1}{n} \sum_i x_i. \tag{4.2}$$

Now suppose we want to compute the subjective probability of some proposition p given that situation X is the case (in fact, p itself can be a combination of propositions). This is the belief value of p in situation X, denoted $\tau(p|X)$. From the fact that $\Pr(p|X) = \Pr(p \wedge X) / \Pr(X)$ and Equations 4.1 and 4.2 it follows that

$$\tau(p|X) = \frac{\sum_i \mu_i(p) x_i}{\sum_i x_i}. \tag{4.3}$$

Note that the belief value of p given a situation X that equals p can be somewhat less than 1, reflecting the uncertainty inherent in fuzzy logic systems. In practice, however, the subjective probabilities correspond very closely to the actual probabilities in the microworld, as is shown in the results of Section 4.5.1.

4.3 Temporal world knowledge

Apart from the important difference in representation, the DSS model and the Golden and Rumelhart model have identical architectures. Be reminded that world knowledge implementation in the Golden and Rumelhart model was based on four simplifying assumptions (see Section 3.1.2):

- World knowledge is implemented as the influence the beliefs in two propositions have on each other.
- World knowledge does not change over story time.
- The belief in propositions at time step t is only influenced by the belief in those at $t-1$ and $t+1$.
- World knowledge is not directed in time.

Of these, only the first assumption does not apply to the DSS model. Instead, it is assumed that world knowledge concerns influences between 'situation-space dimensions' or 'SOM cells'. The nodes of the network in Figure 3.1 (on page 95), showing the architecture of the Golden and Rumelhart model, no longer refer to propositions-at-time-steps but to SOM-cells-at-time-steps. This does not make a difference to the network and, therefore, not to the Markov random field analysis either. The propositional indices $p, q, \ldots$ used in the equations of the Golden and Rumelhart model are simply replaced by indices $i, j, \ldots$ denoting situation space dimensions.

4.3.1 Computing the world knowledge matrix

Since DSS shares its mathematical basis with the Golden and Rumelhart model, knowledge about temporal relations is again implemented in an $n \times n$ world knowledge matrix W. The values in this matrix are based on the temporal contingencies between consecutive situations of the microworld description after converting them into the distributed representation developed in Section 4.2. Unlike the membership values of SOM-cells, the values in W are not obtained by a training procedure. Instead, they are computed directly from the example situations.

Let $S_1, S_2, \ldots, S_{250}$ be the sequence of example situations discussed in Section 4.1. The kth example can be represented by a vector $\mu(S_k)$ in distributed situation space by applying the rules in Equation 4.1. Before W can be computed from these vectors, they need to be normalized by replacing each vector value by the proportional deviation from the vector's average $\mu.(S_k) = \frac{1}{n}\sum_j \mu_j(S_k)$:

$$\nu_i(S_k) = \frac{\mu_i(S_k) - \mu.(S_k)}{\mu.(S_k)}.$$

Each entry in W is computed from these normalized vectors ν:

$$w_{ij} = \frac{1}{K-1}\sum_{k=1}^{K-1} \nu_i(S_k)\nu_j(S_{k+1})$$

where $K = 250$, the number of training situations. If it often happens that two SOM cells i and j both have a high value or both have a low value in consecutive example situations, then w_{ij} will become positive. If i and j often have dissimilar values in consecutive examples, w_{ij} will become negative. In this way, w_{ij} reflects the temporal contingencies between cells i and j.

4.3.2 Belief values in a neighboring situation

Equation 4.3 is used to compute the belief value of a proposition, given the situation at the same time step. We would also like to find an expression for the belief value of proposition p at time step t, given the situations at the neighboring time steps $t-1$ and $t+1$. For this purpose, the temporal world knowledge in W is used to compute a likely situation vector at t from the vectors X_{t-1} and X_{t+1}.

It follows from Markov random field theory (see Appendix B.2) that the most likely value for any SOM-cell i at time step t is either $x_{i,t} = 0$ or $x_{i,t} = 1$. However, we run the risk of being quite wrong if we always choose for one of those values since there may not be much difference between the probabilities of these extremes. To take this into account, it is safer to use the *expected value* of $x_{i,t}$, given X_{t-1} and X_{t+1}. As proven in Appendix B.2, this expected value equals

$$E_{i,t} = \begin{cases} \left(1 - \mathrm{e}^{-\Delta Q_{i,t}}\right)^{-1} - \frac{1}{\Delta Q_{i,t}}, & \text{if } \Delta Q_{i,t} \neq 0 \\ \frac{1}{2} & \text{if } \Delta Q_{i,t} = 0 \end{cases} \tag{4.4}$$

with

$$\Delta Q_{i,t} = X_{t-1}W_{\cdot i} + W_{i\cdot}X'_{t+1}. \tag{4.5}$$

Here, $W_{\cdot i}$ is the ith column of W and $W_{i\cdot}$ is its ith row. The column vector X'_{t+1} is the transpose of the row vector X_{t+1}. In case there is no previous or next situation, so $t = 1$ or $t = T$, it is defined that $X_0 = \vec{0}$ or $X_{T+1} = \vec{0}$, respectively.

Note the resemblance to Golden and Rumelhart's Equations 3.4 and 3.5 that compute the probability of a proposition given the situation(s) at the previous and/or following time step. This similarity does not come unexpectedly since that model is also based on Markov random field theory. There are only two differences between the computation of $E_{i,t}$ by DSS (Equation 4.4) and $\Pr(p_t)$ by Golden and Rumelhart (Equation 3.4). First, the DSS model does not need a bias vector B because a proposition's unconditional belief value follows from its SOM representation according to Equation 4.2. Therefore, B does not appear in Equation 4.5. Second, $\Pr(p_t)$ applies the logistic sigmoidal function to $\Delta Q_{p,t}$, while $E_{i,t}$ applies an alternative sigmoidal to $\Delta Q_{i,t}$. As explained in Appendix B.2, this is a mathematical consequence of the fact that propositions in the Golden and Rumelhart model are either true or false, while SOM cells have real values between 0 and 1.

The *expected situation vector* at time step t is formed by the collection of expected values: $E_t = (E_{1,t}, \ldots, E_{n,t})$. Belief values at time step t, given the situation(s) at $t - 1$ and/or $t + 1$, are computed according to Equation 4.3, using the expected vector E_t instead of an actual situation vector:

$$\tau(p_t|X_{t-1}, X_{t+1}) = \frac{\sum_i \mu_i(p) E_{i,t}}{\sum_i E_{i,t}}. \tag{4.6}$$

4.3.3 Proposition fit and story coherence

Apart from belief values, other measures that can be interpreted psychologically are needed. Two of these are *proposition fit* and *story coherence*, which give information about the temporal relatedness of situations.

Magliano, Zwaan, and Graesser (1999) present data showing that the extent to which a sentence fits in a story context is rated higher if the sentence is more causally connected to the other story statements. Likewise, we define proposition fit as the proposition's strength of relation with neighboring situations. Suppose proposition p can be expected to occur at time step t given the previous and next situations, for instance because X_{t-1} is a possible cause of p_t, and X_{t+1} is a possible consequence. In that case, the belief value of p_t given X_{t-1}

and X_{t+1} will be larger than the unconditional belief value $\tau(p)$. The difference between the two is the proposition fit of p_t:

$$\text{prop.fit}(p_t) = \tau(p_t|X_{t-1}, X_{t+1}) - \tau(p). \tag{4.7}$$

Story coherence is a measure for the extent to which a sequence of situations is in concordance with temporal world knowledge. If situations X_{t-1} and X_{t+1} increase the amount of belief in the intermediate situation X_t, then the trajectory $\langle X_{t-1}, X_t, X_{t+1}\rangle$ is temporally coherent. The coherence of a complete story trajectory is the increase in belief value by neighboring situations, averaged over all time steps:

$$\text{coh}(\bar{X}) = \frac{1}{T}\sum_t \left(\tau(X_t|X_{t-1}, X_{t+1}) - \tau(X_t)\right). \tag{4.8}$$

4.4 The inference process

The statements of a story, together with world knowledge, put constraints on the propositions that can be the case in the story. According to the DSS model, inference is reflected in the propositions' belief values changing to satisfy these constraints. This means that propositions that are likely to be the case in the story have their belief values increased, while the belief values of unlikely propositions are decreased.

Before running the model on a story, the story's situations are converted into vectors in situation space, as explained in Section 4.2. Next, the model starts processing on the first two situation vectors X_1 and X_2 simultaneously because no temporal inferences are possible with only a single situation. Contrary to the Golden and Rumelhart model, all the following story situations enter the model one by one, and are processed as they come in. When the inference process is completed for the story so far, the next situation (if any) enters the model and the process resumes. Processing of the older situations resumes as well, so existing inferences about earlier time steps can be withdrawn and new inferences can be made.

During the inference process, the model uses temporal world knowledge to convert the sequence of situation vectors (i.e., the trajectory) corresponding to the story read so far, into one that contains the information present in the story as well as new information inferred from it. This means that the process needs to solve two problems. First, the facts given by the story should be preserved. Second, the story trajectory should be adapted to temporal world knowledge.

4.4.1 Preventing inconsistency

Preventing text-given propositions from being denied is straightforward in Golden and Rumelhart's localist situation space: These propositions are never allowed to have their belief values changed. In the DSS model, there is no direct connection between situation vectors and propositions. Still, it is not difficult to prevent conclusions inconsistent with the original story. Everything outside a story situation's SOM area belongs to the negation of the situation and may therefore not be inferred during the inference process. A story situation plus extra information is always a subarea of the original situation's area. If $x_{i,t}(0)$ is the value of SOM-cell i at time step t of the original story's trajectory, then after any amount of processing time, the current value $x_{i,t}$ may not be larger than $x_{i,t}(0)$. Therefore, setting a maximum value $x_{i,t}^{\max} = x_{i,t}(0)$ for each SOM cell prevents inferences that are inconsistent with the statements given by the text.

4.4.2 Applying temporal knowledge

Knowledge about the temporal patterns occurring in the microworld is encoded in matrix W. During the inference process, the trajectory is brought into closer correspondence with this matrix. This temporal pattern matching is accomplished by adjusting all individual values $x_{i,t}$ towards levels that are more likely considering the current trajectory and the values in W.

Equation 4.4 gives a cell's expected value $E_{i,t}$, given the rest of the trajectory and world knowledge. If this value is larger than .5, the most likely value is $x_{i,t} = 1$, so $x_{i,t}$ has to increase (taking into account that its maximum value is $x_{i,t}^{\max}$). When $E_{i,t}$ is smaller than .5, the most likely value is $x_{i,t} = 0$, so $x_{i,t}$ should decrease (taking into account that it cannot become negative). This is done for all values in the trajectory $\bar{X}$ in parallel.

Since the expected values at time step t depend on the current values of $X_{t\pm1}$, and the expected values at $t \pm 1$ depend on X_t, the xs cannot be set to 0 or $x_{i,t}^{\max}$ directly. In the models of Chapter 2, as in Golden and Rumelhart's model, values that needed to change gradually were updated iteratively over a number of discrete processing cycles. The DSS model, on the other hand, acknowledges that processing time is continuous. Its process is not defined by an equation stating how the trajectory at cycle $c + 1$ is computed from the one at c. Instead, a first-order differential equation states how the *change* in value of $x_{i,t}$ over processing time, denoted $\dot{x}_{i,t}$, depends on the current trajectory. Given an initial trajectory, this equation can be solved approximately, giving the development of the trajectory over continuous time expressed in arbitrary 'model processing time' units. The vectors $X_t(0)$ of the original story trajectory serve as the initial values for this evaluation. The equation is solved by the function ODE45 in MATLAB 6.1, using a method developed by Dormand and Prince (1980).

The DSS model's inference process is defined by

$$\dot{x}_{i,t} = \begin{cases} \left(E_{i,t} - \frac{1}{2}\right)\left(x_{i,t}^{\max} - x_{i,t}\right) & \text{if } E_{i,t} > \frac{1}{2} \\ \left(E_{i,t} - \frac{1}{2}\right) x_{i,t} & \text{if } E_{i,t} \leq \frac{1}{2}. \end{cases} \tag{4.9}$$

The factor $(E_{i,t} - \frac{1}{2})$ makes sure that $x_{i,t}$ always changes towards a more likely value: It increases as long as $E_{i,t} > .5$ and decreases when $E_{i,t} < .5$. If $x_{i,t}$ is increasing, its rate of change is multiplied by its distance to the maximum value $x_{i,t}^{\max}$, preventing $x_{i,t}$ from becoming larger than this maximum. If $x_{i,t}$ is decreasing, its rate of change is multiplied by its distance to 0, preventing $x_{i,t}$ from becoming negative.

Note that the inference process depends only on the values of the SOM cells and the world knowledge matrix W. Psychologically interpretable measures such as story coherence and belief values do not control or even influence the process, but only reflect its outcome.

4.4.3 Depth of processing

When a situation has been sufficiently processed, the next situation is allowed to enter the model. The criterion for sufficient processing is controlled by a positive depth-of-processing parameter θ. Equation 4.9 is evaluated until the trajectory's total rate of change is less than the threshold value $1/\theta$:

$$\sum_{i,t} |\dot{x}_{i,t}| < \frac{1}{\theta} \tag{4.10}$$

where t ranges from 1 to the number of story situations in the model at that moment. Large values of θ correspond to deep processing, since story situations are added when inferencing on the previous situations is mostly completed. As θ decreases, the criterium for convergence becomes less stringent and the process halts even if much can still be inferred, corresponding to shallower processing. In all simulations presented here, the value of θ was set to 0.3 unless stated otherwise. In Appendix B.3 it is proven that the stopping criterium is always reached eventually.

4.4.4 Amount of inference

At any moment during the inference process, the trajectory can be interpreted by computing the belief value of any proposition-at-time-step p_t. If the process results in an increase of belief in p_t, this means that p_t is positively inferred. Likewise, if its belief value decreases, p_t is negatively inferred, meaning that it is inferred not to be the case. Formally, the amount of inference is defined as the increase of the proposition's belief value relative to its value in the original story:

$$\mathrm{inf}(p_t) = \tau(p_t|X_t) - \tau(p_t|X_t(0)). \tag{4.11}$$

Also, we require a measure for the total amount of inference that takes place during processing. This is not simply Equation 4.11 summed over all basic propositions, because complex propositions should be taken into account as well. The total amount of inference is therefore determined by directly comparing the initial story trajectory to the result of its interpretation.

The unconditional belief value of a story situation $X_t(0)$ is computed using Equation 4.2. When new facts about the situation at time step t are inferred, it is replaced by a more informative situation X_t. Since this new situation is more specific, it is less likely to occur and has a lower unconditional belief value. The total amount of inference on the situation at time step t equals its decrease in unconditional belief value: $\tau(X_t(0)) - \tau(X_t)$. This can be interpreted as the increase in the amount of knowledge there is about the situation. The total amount of inference on a trajectory is the sum of the amounts of inference on its individual situations:

$$\text{total inf}(\bar{X}) = \sum_t (\tau(X_t(0)) - \tau(X_t)) \tag{4.12}$$

where t ranges from 1 to the number of story situations in the model at that moment. Note that the total amount of inference is largest when $\tau(X_t) = 0$, which is only the case if X_t equals the nilvector. If this happens, the model has inferred that the situation was inconsistent with the rest of the story and should not be believed. A reader making such an inference may well discard it and accept the story at face value, awaiting further information and resulting in no inference made. The model does not include a process that evaluates its inferences. Therefore, when faced with a sequence of situations that is inconsistent according to world knowledge, it can be inferred that one of the situations is impossible.

4.5 Results

4.5.1 World knowledge implementation

How can we ascertain that world knowledge was implemented successfully in the situation space and world knowledge matrix W? Be reminded that belief values, based on this world knowledge implementation, can be interpreted as the subjective probabilities of propositions. The probabilities also follow directly from the microworld description. By comparing these 'actual' probabilities (Pr) to the belief values (τ), it can be established whether the model's world knowledge reflects the regularities that hold in the microworld.

First, the subjective and actual probabilities of all conjunctions $p \wedge q$ of (negations of) basic propositions were compared. Since p and q may denote the same proposition, this includes all single basic propositions. The resulting scatter plot is shown in the left panel of Figure 4.4.

Second, we tested whether the non-temporal dependencies among propositions are captured by their vector representations. If a (negation of a) basic proposition p is given, the probability that a positive basic proposition s is the case at the same moment in the microworld description changes by an amount $\mathrm{Pr}(s|p) - \mathrm{Pr}(s)$. The center panel of Figure 4.4 shows the scatter plot of these actual probability differences versus their corresponding subjective probability differences.

Third, we tested whether the world knowledge matrix W correctly captures temporal dependencies among propositions. In the microworld, the amount of influence that a proposition p at time step $t \pm 1$ has on a proposition s at t is $\mathrm{Pr}(s_t|p_{t\pm1}) - \mathrm{Pr}(s)$, the change in probability of s_t. All positive basic propositions s_t and (negations of) basic propositions $p_{t\pm1}$ were used for the scatter plot of actual versus subjective probability differences in the right panel of Figure 4.4.

In all three cases, the correlation between actual and subjective probabilities was very high: .996, .979, and .914, respectively. Also, the three scatter plots show that there are no outliers. In short, the vector representation of propositions and the world knowledge matrix W did capture the regularities that occurred in the microworld description.

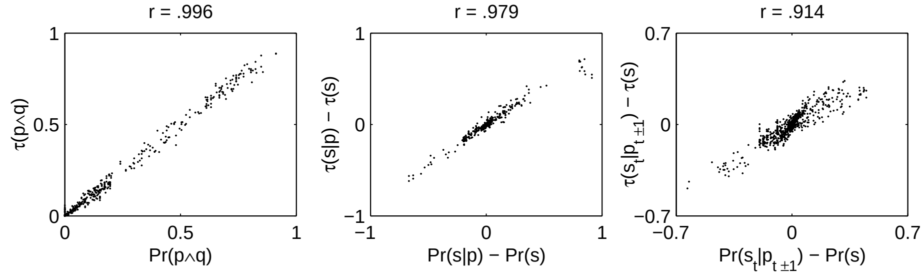

Figure 4.4: Scatter plots of actual probabilities (Pr) versus subjective probabilities (τ), and coefficients of correlation (r). Propositions denoted by p or by q are basic propositions or negations thereof. Propositions denoted by s are positive basic propositions.

4.5.2 Specific inferences

In order to test the model's ability to make specific inferences, three simple sequences of situations ('stories') were constructed. These stories, shown in Table 4.2, varied in length from two to nine situations. Each story was meant to evoke one or more specific inferences.

Story 1: Realizing the exclusive-or relation

From the fact that Bob or Jilly wins, it can be inferred that they must have been at the same place in the previous time step. This inference requires the exclusive-or relation: If *either* Bob *or* Jilly is outside at t, winning cannot occur at $t+1$; if *both* are (not) outside at t, winning is possible at $t+1$. Story 1 tests whether this knowledge was successfully implemented in matrix W. The situation at $t = 1$ gives no indication where Bob and Jilly might be. Following this, someone wins, which means that they both must have been either outside or not outside (which is equivalent to being inside) at $t = 1$. The model is able to correctly infer this, as can be seen from Figure 4.5. This result shows that the DSS model can handle the exclusive-or relation required to make this inference.

The amounts of inference of 'Bob and Jilly are outside' and of 'Bob and Jilly are inside' seem fairly low. There are two reasons for this. First, these two situations exclude each other and can therefore never be both strongly inferred. Second, Bob and Jilly are a priori more likely to be at the same place than to be at different places. As a result, the belief values for 'Bob and Jilly are (not) outside' are high to begin with and cannot increase much more.

Table 4.2: Three stories used to model specific inferences. For each situation is shown how it is constructed from basic propositions and a possible text describing this situation is given.

story	t	situation	possible text
1	1	¬RAIN ∧ ¬SUN	*It doesn't rain and the sun doesn't shine.*
	2	(SOCCER ∨ HIDE-AND-SEEK ∨ (B COMPUTER ∧ J COMPUTER)) ∧ (B WINS ∨ J WINS)	*Bob and Jilly are playing a game and one of them wins.*
2	1	SUN	*The sun is shining.*
	2	HIDE-AND-SEEK	*Bob and Jilly are playing hide-and-seek.*
	3	¬(B OUTSIDE) ∧ ¬(J OUTSIDE)	*They are inside.*
3	1	SUN ∧ SOCCER	*The sun shines and Bob and Jilly play soccer.*
	2	B TIRED ∧ ¬(J TIRED)	*Bob is tired, but Jilly isn't.*
	3	B WINS ∨ J WINS	*Next, one of them wins.*
	4	B TIRED ∧ J TIRED	*Now they are both tired.*
	5	RAIN	*It starts raining.*
	6	B INSIDE ∧ J INSIDE ∧ HIDE-AND-SEEK	*Bob and Jilly go and play hide-and-seek inside.*
	7	J TIRED ∧ ¬(B TIRED)	*Only Jilly is tired.*
	8	B WINS ∨ J WINS	*Someone wins.*
	9	B COMPUTER ∧ J DOG	*Later, Bob is playing a computer game, and Jilly is playing with the dog.*

Story 2: Retracting an inference

After reading the first two sentences of Story 2, one might infer that Bob and Jilly play hide-and-seek *outside*. This inference is based on the information that the sun shines and on the knowledge that this usually causes them to be outside. However, the third sentence tells us that they are in fact inside at $t = 3$. This does not necessarily mean that they were already inside at $t = 2$, but it does make that more likely. Therefore, the inference that Bob and Jilly are outside at $t = 2$ should be retracted. As Figure 4.6 shows, this is indeed what the model does. At first, the belief value of 'Bob and Jilly are outside' at $t = 2$ increases. After 5.38 units of model processing time have passed, the process stabilizes enough to allow the third situation to be added to the story trajectory. From that moment, the belief value decreases almost to its original level: It is no longer inferred that Bob and Jilly are outside during story time step $t = 2$.

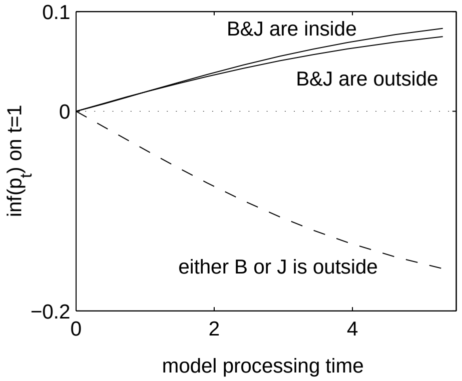

Figure 4.5: Amount of inference (Equation 4.11) of 'Bob and Jilly are outside' (B OUTSIDE $\wedge$ J OUTSIDE), of 'Bob and Jilly are inside' ($\neg$(B OUTSIDE) $\wedge$ $\neg$(J OUTSIDE)), and of 'either Bob or Jilly is outside' (B OUTSIDE XOR J OUTSIDE) during processing of Story 1.

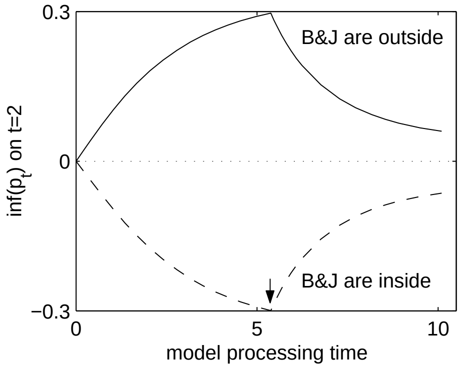

Figure 4.6: Amount of inference (Equation 4.11) of 'Bob and Jilly are outside' and of 'Bob and Jilly are inside' at $t = 2$, during processing of Story 2. The third situation enters the model after 5.38 units of processing time, as indicated by the arrow.

Story 3: Inferring who wins at what

Whoever is tired, is less likely to win. In Story 3, it is Bob who is tired at first, so the one who wins at $t = 3$ is probably not him, but Jilly. The left graph in Figure 4.7 shows that the model infers exactly this. The right graph shows that Bob is inferred to win later in the story ($t = 8$), when Jilly is tired.

Also, the model infers what Bob and Jilly are playing when one of them wins. Note that the game being played is mentioned two time steps before it is stated that someone wins. Still, it is inferred that the game being won is soccer at $t = 3$ and hide-and-seek at $t = 8$. Since situations are only directly influenced by the previous and next time steps, this information must have travelled through the intermediate time steps $t = 2$ and $t = 7$ respectively, showing that indirect influence from more distant situations is indeed possible.

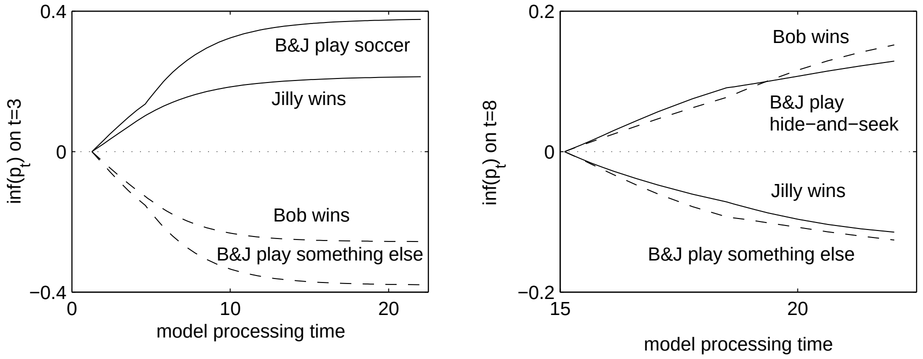

Figure 4.7: Amounts of inference (Equation 4.11) during processing of Story 3. Left: inference at $t = 3$ of 'Bob wins', of 'Jilly wins', of 'Bob and Jilly play soccer', and of 'Bob and Jilly play something else' (HIDE-AND-SEEK ∨ (B COMPUTER ∧ J COMPUTER)). The third situation is added to the story trajectory at 1.19 units of processing time after the inference process began with the first two situations. Right: inference at $t = 8$ of 'Bob wins', of 'Jilly wins', of 'Bob and Jilly play hide-and-seek' and of 'Bob and Jilly play something else' (SOCCER ∨ (B COMPUTER ∧ J COMPUTER)). The eighth situation enters the model at 15.10 units of processing time.

4.5.3 Inferences in general

The previous section shows that the model's inferences correspond to our intuitions: Propositions that are implied by the story are inferred. In order to test this more systematically, 100 random stories were constructed and used as input to the model. The stories varied in length from three to seven situations. There were 20 random stories of each length, so the total number of story situations equaled $20 \times (3 + 4 + 5 + 6 + 7) = 500$. Each such situation consisted of exactly one basic proposition or its negation.

In general, propositions that are inferred on the basis of temporal world knowledge should be

- implied by the story;
- possible given the story situation. If it is stated that Bob is outside, it cannot be inferred that he is inside at the same moment in story time;
- not already given by the story situation. If Bob and Jilly play soccer, then they must be outside. This inference does not require information from other story situations, and is therefore not an inference in the sense of Equation 4.11.

For each situation of each random story, the proposition fit (Equation 4.7) and amount of inference (Equation 4.11) of all basic propositions were obtained (except for the proposition that constituted the situation). The correlation between amount of inference and fit of propositions was .66 (based on 500 situations $\times(14-1)$ propositions = 6,500 observations), indicating that the model does indeed infer propositions that are implied. Moreover, propositions with positive fit were inferred to be the case (positive inference) and propositions with negative fit were inferred to be not the case (negative inference).

Whether a proposition p_t is possible given the original story situation $X_t(0)$ can be seen from its initial belief value $\tau(p_t|X_t(0))$. If this value is close to 0, p_t is unlikely to be the case at that moment in the story and should not be inferred even if it is a likely proposition given the rest of the story. Likewise, if the initial belief value is close to 1, p_t is already likely given story situation $X_t(0)$ and it should not be inferred to be the case at t from situations at other story time steps.

Indeed, this is what the model predicts. All 500 situations × 14 basic propositions = 7,000 observations were divided into two groups. The 'non-inferable' group contained cases with initial belief values so close to 0 or 1 (less than .001 or more than .999) that inference was not expected to occur. The 'inferable' group contained the others. The average absolute proposition fit was .11 among the non-inferables and .08 among the inferables, indicating that the latter would be inferred less if only proposition fit would matter. However, the opposite was the case: The average absolute amount of inference was .07 among the inferables but only 2.4×10^{-5} among the non-inferables.

4.5.4 Inference and coherence

It is generally assumed that the inferences readers most easily make on-line are inferences that contribute to the coherence of the story, which in the model is defined in Equation 4.8. The coherences of the 100 random stories ranged from $-.23$ to .25, with an average of .001. Since coherence is a measure for the match between a story and temporal world knowledge, and the inference process adapts the trajectory to world knowledge matrix W, the story coherences of the trajectories increased through this process. The result was a larger coherence value for all 100 stories (the average was .28), showing that the inferences contributed to the stories' coherence. However, this is not a built-in consequence of the model's equations: Transient decreases of coherence during processing were observed for 17 stories, taking 4.5% of their processing time.

4.5.5 Relatedness, inference and reading time

A story sentence is read faster when it is more related to the preceding sentence. Myers, Shinjo, and Duffy (1987), and also Golding, Millis, Hauselt, and Sego (1995), showed this by having subjects read stories consisting of just two events. The relatedness between those events varied: The second story event was either unrelated to the first event or was predictable to a certain degree. They found that reading the second sentence took more time when it was less related to the first sentence. Murray (1995, 1997) also had subjects read two-sentence stories but included stories in which the events were adversatively related, meaning that the first story event made the second event less likely to occur. He found that the second sentence took more time to read when it was adversatively related to the first sentence than when it was unrelated. Using more realistic texts, Sanders and Noordman (2000) showed that a sentence is read faster when it is embedded in a text that causally implies it, than when it is not causally related to the rest of the text.

To test whether the model predicts the same relation between relatedness and reading time, five stories with different levels of relatedness were constructed. Each of the stories, shown in Table 4.3, consisted of three situations, the first of which was 'Bob and Jilly play soccer' and the last was 'Bob wins'. Relatedness was varied among stories by modifying the second situation. Since Bob is more likely to win when Jilly is tired, stating that Jilly is tired and Bob is not, should result in the highest relatedness to the last situation. If, on the other hand, Bob is tired and Jilly is not, relatedness is lowest. Intermediate levels of relatedness are obtained in a similar way.

Table 4.3: Stories with different relatedness levels. Relatedness level is varied by using one of the five situations at $t = 2$.

t	relatedness level	situation	possible text
1		SOCCER	*Bob and Jilly play soccer.*
2	1	B TIRED $\wedge$ $\neg$(J TIRED)	*Only Bob is tired.*
	2	B TIRED	*Bob is tired.*
	3	B TIRED XOR J TIRED	*One of them is tired.*
	4	J TIRED	*Jilly is tired.*
	5	J TIRED $\wedge$ $\neg$(B TIRED)	*Only Jilly is tired.*
3		B WINS	*Bob wins.*

The time needed by the model to process the last situation and the amount of inference that took place during this process are plotted in Figure 4.8. These results clearly show that a higher level of relatedness leads to shorter processing time and less inference, which is consistent with the generally accepted idea that the on-line construction of an inference takes time. For instance, Vonk and Noordman (1990) had subjects read texts that contained an inference evoking sentence. When the information to be inferred was explicitly stated in the text before the inferring sentence, reading times on the inferring sentence were shorter than when the information was not stated but had to be inferred.

Stories describing less related events evoke more inferences, which slows down reading. To test whether this relation holds in general, the model was run on all stories consisting of just two situations, with each situation consisting of exactly one (negation of a) basic proposition. Since there are 14 positive basic propositions, the number of stories was $(2 \times 14)^2 = 784$. Story coherence was taken as a measure of relatedness. The coherence values ranged from $-.34$ to .38, so the relatedness of the two situations ranged from adversative to predictable.

Figure 4.9 directly compares the model's results to those of Myers et al. (1987) and Golding et al. (1995). Since stories with adversatively related sentences were not used in those studies, only the model's results for the 394 stories with nonnegative coherence are plotted. The effect of story coherence on pro-

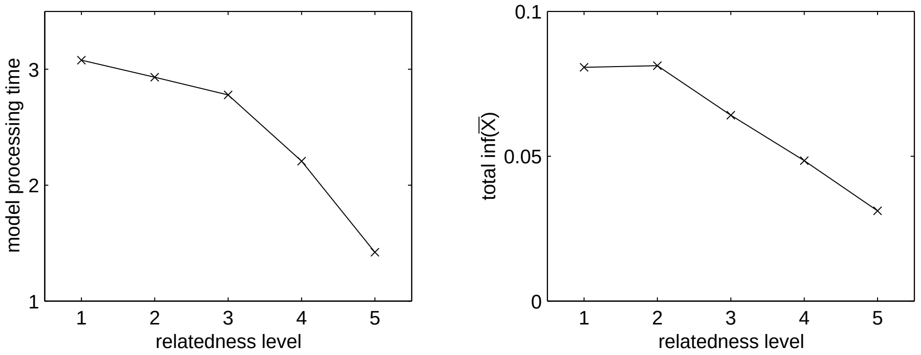

Figure 4.8: Amount of time needed to process the situation 'Bob wins' (left) in the stories of Table 4.3, and total amount of inference (Equation 4.12) that took place during this process (right), as functions of the situation's relatedness to the previous situation.

cessing time as found by the model is quite similar to the effect of relatedness on reading time as found by Myers et al. and Golding et al.

Over all 784 stories tested, the correlation between story coherence and amount of inference was $-.42$. A sequence of story situations that violates

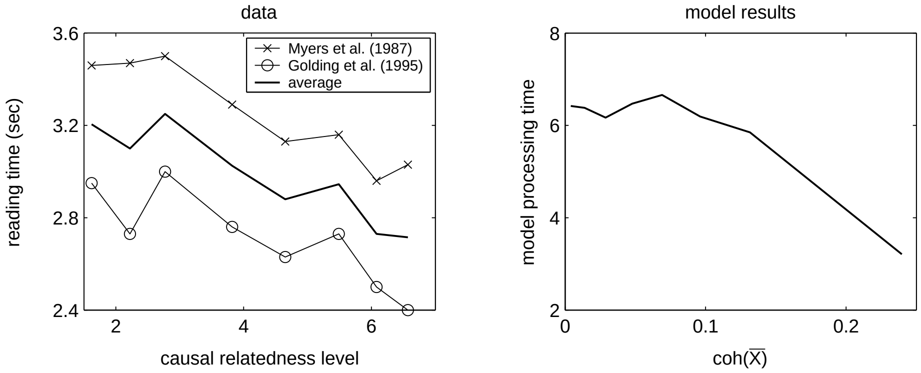

Figure 4.9: Left: reading time on the second sentence of two-sentence stories, as a function of sentence relatedness (Myers et al., 1987, Figure 1; Golding et al., 1995, Figure 7.1). Right: amount of processing time needed to process two-sentence stories, as a function of story coherence (Equation 4.8). Each of the eight points in the graph is the average of processing times and coherences for 49 or 50 stories.

temporal world knowledge will evoke more inferences than a story that is in accordance with world knowledge. This results in an increase in processing time. Accordingly, there was a negative correlation ($r = -.36$) between story coherence and model processing time. The model correctly predicts that stories with adversatively related events are processed more slowly than stories that describe unrelated events, and that stories describing positively related events are processed quickest. The strong relation between amount of inference and processing time was also reflected in the high positive correlation ($r = .93$) between the two.

4.5.6 Inference and depth of processing

Noordman, Vonk, and Kempff (1992) varied subjects' reading goal by either instructing them to check for inconsistencies in a text, or by not giving such an instruction. They found that the consistency-checking instruction led to more inferences and longer reading times. Stewart, Pickering, and Sanford (2000) used another method to manipulate the reading process. Their subjects read single sentences and had to answer a related question after every sentence. In one condition, all of these questions could be answered without making any inference from the sentences, while in the other condition inferences needed to be made from every sentence. It was found that reading slowed down when inferencing was required compared to when it was not.

Supposedly, instructing readers to check for inconsistencies or having them answer inference-requiring questions leads to deeper processing of the texts. In the model, depth of processing is controlled by parameter θ. Figure 4.10 shows the effect of varying θ on average processing time per situation and total amount of inference during processing of each situation, for the 100 random stories. In accordance with empirical data, deeper processing resulted in longer processing times[3] and more inference.

[3] From Equation 4.10, which determines when processing of a situation is completed, it might seem as if processing time can never decrease with increasing θ. However, this only is true for the first two story situations. Deeper processing can lead to shorter processing time for the situation at $t+1$ if this situation is highly compatible with an inference that was made at story time step t. With shallower processing, this '$t+1$-compatible' inference might not be made, leading to longer processing time for $t+1$. In fact, when comparing $\theta = 0.3$ to $\theta = 0.6$, this effect occurs in three of the 100 random stories.

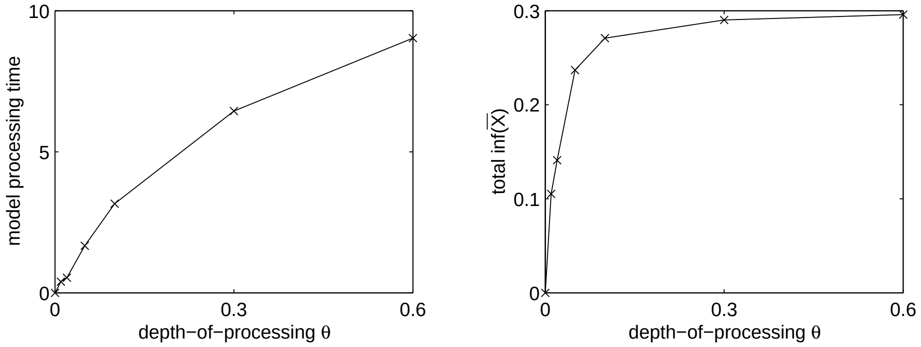

Figure 4.10: Effect of depth-of-processing parameter θ on average processing time per situation (left) and total amount of inference (Equation 4.12) during processing of each situation (right).

4.6 Conclusion

The DSS model takes as input a story in the form of a temporal sequence of situations and uses world knowledge about temporal contingencies to infer which propositions not stated in the story are likely to be the case nevertheless. So far, it does not offer more than does the Golden and Rumelhart model or the Story Gestalt model. Like those models, the DSS model is computationally sound, which comes as no surprise since it shares its mathematical foundations with the Golden and Rumelhart model. Considering Jacobs and Grainger's (1994) four evaluation criteria, however, the DSS model can be said to be an improvement over the two comparable models.

4.6.1 Simplicity

Considering the size and complexity of the equations involved in the DSS model, it seems at first not to score well on simplicity. However, simplicity should not be taken to mean 'ease of understanding'. The model is, in fact, quite simple in the sense that it has just one free parameter and all its equations follow from only four simplifying assumptions.

The model's only free parameter, θ, can be interpreted psychologically as controlling depth of processing, and simulation results showed support for such an interpretation. The four architectural assumptions concerning the implementation of temporal world knowledge (Section 4.3) lead to the Markov random field (MRF) analysis. Together with the self-organized vector representation of propositions, this yields definitions of belief values which were shown to correspond closely to probabilities in the microworld. These belief values form the basis for measures such as amount of inference, proposition fit, and story coherence. In short, the model's foundations are formed by the four assumptions and the organization of the situation space. All other aspects of the model follow directly from these.

The four assumptions were made in order to simplify the MRF analysis, but it is unclear to what extent they limit the model's abilities. In particular, the symmetry assumption claims that the temporal knowledge matrix W can be used to reason forwards in story time and the transposed matrix W' to reason backwards. However, there is no reason to assume that the real world shows the same symmetry.[4]

The high correlation between probability differences ($\Pr(p_t|q_{t\pm1}) - \Pr(p)$) in the microworld and belief value differences ($\tau(p_t|q_{t\pm1}) - \tau(p)$) in the model shows that, at least for this microworld, the symmetry assumption does not seriously limit the quality of the knowledge matrix. In other (micro)worlds, this might be different. Fortunately, the MRF approach is not necessary for the model's functioning and can easily be replaced by only changing the definition of the $E_{i,t}$-function (Equation 4.4). This function gives the expected value for $x_{i,t}$, given the previous and/or next situations and world knowledge. It is the model's central function, since it states how world knowledge is implemented and applied to a story representation. If a better implementation of world knowledge is found, or a better way to apply it to the story trajectory, only the $E_{i,t}$-function needs to be changed accordingly. All of the four assumptions on which world knowledge implementation is based can be discarded if a better $E_{i,t}$-function is to be found without them.

4.6.2 Descriptive adequacy

The model was validated against several empirical findings. Processing of less coherent stories took more time because these stories evoked more inferences than did more coherent stories. Also, increasing depth of processing led to more inference and slower reading. Be reminded that neither the Story Gestalt model nor the Golden and Rumelhart model could predict reading times. The Story Gestalt model does not have the necessary notion of processing time, and the Golden and Rumelhart model does not process story situations separately.

The correspondence between model results and experimental data is not trivial. The model was not designed to predict any particular empirical data but only to perform inferencing by adjusting incomplete descriptions of story events to world knowledge, which it did successfully. Accounting for empirical data is therefore an emergent property of the model.

[4] It is important to note that from the symmetry assumption it does not follow that the belief values are symmetrical. In general, $\tau(p_t|q_{t\pm1}) \neq \tau(q_{t\pm1}|p_t)$. To give an example: If Bob or Jilly wins at t, it is certain that they did not play with the dog at $t-1$. This is reflected in the high belief value $\tau(\neg(\text{B}\vee\text{J DOG})_{t-1}|\ \text{B}\vee\text{J WINS}_t) = .90$. On the other hand, given that Bob and Jilly do not play with the dog at $t-1$, it is not at all certain that one of them will win at t. Indeed, the corresponding belief value is $\tau(\text{B}\vee\text{J WINS}_t|\neg(\text{B}\vee\text{J DOG})_{t-1}) = .24$.

4.6.3 Explanatory adequacy

The two major theories of on-line inference are the minimalist theory (McKoon & Ratcliff, 1992), which claims that readers do not commonly create elaborate situation models during reading, and the constructionist theory (Graesser, Singer, & Trabasso, 1994), which says that readers do form such situation models. Clearly, the DSS model leans more towards the latter account since all of its inferences are based on a situational representation. However, the model also differs from the constructionist theory in one important respect. Graesser et al. claim that readers actively try to accomplish coherence of a text, according to the so-called search-after-meaning principle. In other words, inferencing is driven by a need for coherence. The model offers a reverse interpretation: Increased coherence results from inferences, which emerge from matching the events described in the story to patterns of events known to occur in the world. There is no search for story coherence. Rather, incoming information automatically adjusts the story trajectory, generally resulting in increased coherence.

Technically, making the model coherence-driven is not hard to do. A standard gradient-ascent algorithm can be applied to search for a local maximum of story coherence (Equation 4.8) starting with the original story trajectory. However, such a coherence-driven implementation is theoretically excluded in our approach. The definition of story coherence is based on belief values, which depend on the situation vectors but cannot influence them. Therefore, the inference process can never be controlled by the story's coherence. Nevertheless, if the increase of coherence is wrongly interpreted as the driving force of the process instead of its consequence, this leads to the illusion of an active search for coherence. The switch from localist to distributed representations makes clear how belief values and coherence form an abstraction, based on a story representation, and can therefore not change the story representation.

4.6.4 Generality

The smallest unit defined in the Golden and Rumelhart model is the proposition. A complex proposition can only be represented if it is a conjunction of basic propositions. Any other complex proposition that is either stated in the story to be processed, or that might be inferred during story processing, needs to be explicitly present in the model by giving it its own dimension in the localist situation space. However, the logical relation between such a complex proposition and its constituent parts is lost in this localist representation.

In the DSS model, the smallest unit is the SOM cell. All complex story situations can be represented as a vector of cell values after training a Self-Organizing Map using only a few basic propositions. These representations implement knowledge about constraints between propositions within a time step. Such non-temporal world knowledge is not available to the Golden and Rumelhart model. Moreover, the ability to represent complex propositions makes the implementation of more complex temporal contingencies (like the exclusive-or relation) possible. Because of this, the DSS model has higher stimulus generality than the Golden and Rumelhart model.

On the other hand, the Golden and Rumelhart model can be claimed to have higher task generality since it is also used to simulate recall of stories. In the following chapter, it will be shown how the DSS model can easily be extended similarly to simulate story retention. Also, two other extensions to the model are presented.

5
Extending the DSS model

The Distributed Situation Space model presented in the previous chapter, which shall be referred to as the *standard DSS model* from now on, can process any story taking place in the microworld it has knowledge of, making it a model with high stimulus generality. When it comes to task generality, however, it does nothing more than make knowledge-based inferences during story comprehension. If it can be shown that other psycholinguistic phenomena can be simulated using the same architecture, this will greatly increase the model's value.

In this chapter, three extensions to the standard DSS model are presented. First, Section 5.1 describes a model that simulates how stories are forgotten over time. Second, a model for the resolution of ambiguous pronouns is added to standard DSS in Section 5.2. Third, in Section 5.3, the possibility of adding a textbase-level representation is discussed.

5.1 Story retention

Like the story itself, the episodic memory trace that remains after reading the story can be represented as a trajectory through situation space in the DSS model. Over time, this memory trace becomes weaker as the story is forgotten, which can be modeled as a change in the story trajectory. Note that such a retention model is very different from Golden and Rumelhart's recall model discussed in Section 3.2.2, even though the DSS inference model is quite similar to theirs. Golden and Rumelhart model the reconstruction of a story trajectory, based on a memory trace whose strength is controlled by the retention interval parameter. We model the process by which the memory trace weakens. Therefore, the retention interval is not controlled by a parameter in our model, but corresponds to the time over which the model's equation is evaluated.

5.1.1 The retention model

As time elapses since a story was read, the amount of information in the story's memory trace decreases. In DSS, a situation that covers a large part of the SOM contains less information than a situation that covers only a small part. Therefore, reducing the amount of information in a trajectory corresponds to an increase in the SOM-cell values $x_{i,t}$. The rate of increase does not need to be the same for all cells. It can be expected that some cell values are more 'stable' than others if there is much evidence, from the rest of the trajectory and world knowledge, that these cells should have small values. In that case, these values should drift up more slowly.

A cell's expected value $E_{i,t}$, as computed by Equation 4.4, can serve as a measure for the instability of the value $x_{i,t}$. Smaller values of $E_{i,t}$ correspond to more stable cell values $x_{i,t}$, because small $E_{i,t}$ indicates that the rest of the trajectory provides much evidence that $x_{i,t}$ should be small. The weakening of a story's memory trace over retention time is therefore modeled by the differential equation

$$\dot{x}_{i,t} = E_{i,t} \left(1 - x_{i,t}\right). \tag{5.1}$$

Each cell's value $x_{i,t}$ increases at a rate equal to the cell's expected value, so cells with larger $E_{i,t}$ will generally increase in value faster than cells with lower

$E_{i,t}$. By multiplying the rate of increase by the distance from $x_{i,t}$ to 1, cell values are prevented from exceeding this maximum value.

The story trajectory resulting from standard DSS serves as the initial value for the evaluation of Equation 5.1. The time over which the equation is evaluated corresponds to the amount of time that elapsed since the story was read, expressed in arbitrary 'model retention time' units. Note that these are not the same as the 'model processing time' units of the standard model that were used to predict reading time.

The retention process, like the inference process, results in adjusted belief values without being affected by belief values. As retention time grows towards infinity, all SOM-cell values $x_{i,t}$ approach 1. By Equations 4.2 and 4.3 this means that any belief value $\tau(p_t|X_t)$ becomes equal to its unconditional $\tau(p)$. In other words, the belief values of the story's propositions regress to their unconditional levels. The extent to which a proposition is still retained can therefore be defined as the difference between its current and its unconditional belief value:

$$\text{ret}(p_t) = \tau(p_t|X_t) - \tau(p). \tag{5.2}$$

After an infinite amount of retention time, the amount of retention of any proposition equals 0: The story is completely forgotten. Amount of retention is also defined for propositions that were never part of the story, so the model can make predictions about the possibility that propositions are falsely recalled.

5.1.2 Results

The 100 random stories introduced in Section 4.5.3 were processed by the standard DSS model, and the resulting trajectories served as input to the retention model. During the retention model's processing, the trajectories' coherences were recorded, as was the amount of retention of all basic propositions.

Although retention of story propositions decreases as retention time grows (Figure 5.1, left) the average coherence of the retained trajectories shows an *increase* before it starts to decrease after approximately 3 units of retention time (Figure 5.1, right). As retention time grows, all SOM-cell values eventually approach 1, which by Equations 4.6 and 4.8 means that all story coherences equal 0.

There are two explanations for the increase in coherence during retention.

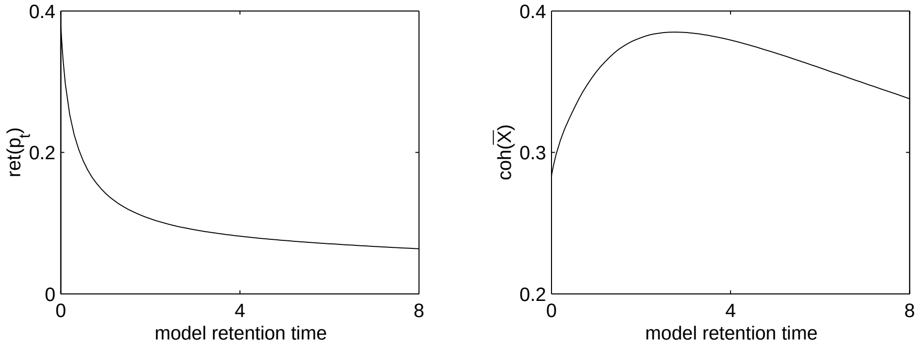

Figure 5.1: Left: average amount of retention (Equation 5.2) of story propositions of the 100 random stories, as a function of retention time. Right: average coherence (Equation 4.8) of retained story trajectories as a function of retention time.

First, propositions may be forgotten selectively. Indeed, it is well known that some story propositions are recalled more easily than others. In a cued recall task, Myers, Shinjo, and Duffy (1987) found that, in general, a sentence was more likely to be recalled if it was more related to the one given as a recall cue. However, the highest levels of relatedness resulted in a small decrease in recall. This last effect was not found by Varnhagen, Morrison, and Everall (1994). They had children read a number of stories and asked them to recall as much of the stories as possible, without giving any sentences as cue. Story propositions with many causal connections in the story were recalled more often than propositions with fewer connections. No decrease in free recall for the highest levels of connectivity was found. The same relation between number of causal connections and recall probability was found by Trabasso and Van den Broek (1985) and by Fletcher and Bloom (1988). In short, propositions that form the 'causal backbone' of the story are remembered best. Moreover, Goldman and Varnhagen (1986) found that this effect is stronger in a delayed free recall task than in immediate recall. Not surprisingly, they also found that fewer story propositions are recalled in delayed recall than in immediate recall, as did many other researchers (e.g., Duffy, Shinjo, & Myers, 1990; Trabasso & Van den Broek, 1985).

Another reason for the increasing coherence might be the occurrence of intrusions. Readers occasionally recall propositions that were never part of the text, but are part of their knowledge. Bower, Black, and Turner (1979) as well as Smith and Graesser (1981) found that propositions that form part of a story script and are therefore highly predictable in the story, are falsely recalled more often than less predictable propositions. Luftig (1982), too, found higher intrusion rates of propositions that, according to world knowledge, follow from the text than of propositions that do not. Moreover, this effect was stronger in a delayed recall task than in immediate recall.

If the model accounts for these empirical data, there should be a positive correlation between proposition fit (Equation 4.7) and retention, both for story propositions and for non-story propositions (intrusions). Moreover, these correlations should increase over retention time. Figure 5.2 shows that the model does predict all of these effects. After 2 units of retention time, the correlation between fit and retention is .49 (based on 500 observations) for story propositions and .58 (based on 6,500 observations) for non-story propositions.

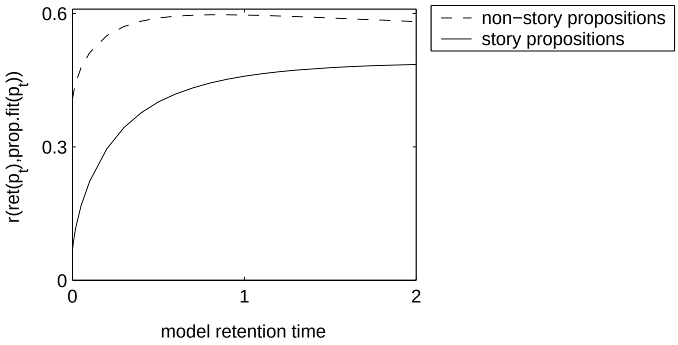

Figure 5.2: Correlation between retention (Equation 5.2) and fit (Equation 4.7) of propositions as a function of model retention time, for story propositions and non-story propositions (intrusions).

5.1.3 Conclusion

Despite its remarkable simplicity, the retention model correctly predicts several empirical findings. First, and least surprisingly, story retention decreases as retention time grows. Second, propositions that are strongly related to the rest of the story are retained better than less related propositions, and this effect grows stronger over retention time. Third, intrusion of propositions that are predictable given the story is more common than intrusion of less predictable propositions. This effect, too, grows stronger over retention time.

The results of the standard DSS model showed that inferences contributed to the stories' coherences. The retention process behaves similarly: Coherence initially increases as the memory trace weakens. Like the inference process, the retention process does not look for coherence nor are propositions in any way selected to be retained or forgotten. Preservation of coherence simply follows as an emergent property from the differential equation that defines the retention process. This equation does not know about coherence or even propositions, and cannot make use of such higher-level concepts. The retention model therefore shows how selective forgetting and intrusion can occur without being caused by a need or search for coherence.

5.2 Pronoun resolution

A paper based on this chapter is submitted for publication as Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003d), *The effect of focus and knowledge on the resolution of ambiguous pronouns: a computational model*. A previous version of the pronoun resolution process was presented in Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003a), A model for knowledge-based pronoun resolution, In: F. Detje, D. Dörner, & H. Schaub (Eds.), *The logic of cognitive systems: Proceedings of the fifth international conference on cognitive modeling* (pp. 245-246), Bamberg, Germany: Universitäts-Verlag.

When comprehending a sentence like *Bob lied to Joe because he could not handle the truth*, several sources of information can be used to find the intended referent of the ambiguous pronoun *he*. One such source of information is the focus on discourse entities, which we take to be the entities' accessibility in the reader's mental representation of the discourse. The more focused an entity, the more likely it is to be chosen as the pronoun's antecedent.

In the example above, the first-mention effect (Gernsbacher & Hargreaves, 1988) causes Bob to be in focus, making it likely that *he* is interpreted to refer to Bob. Focus can also arise from implicit causality, which is the ability of some verbs to bias the interpretation of a following pronoun. The verb *to lie to* is a so-called NP1-biasing verb (or NP1 verb for short), meaning that it biases towards the first noun phrase: The cause of lying is implicitly assigned to the liar and not to the person being lied to. Therefore, after reading *Bob lied to Joe because he...*, the pronoun is commonly taken to refer to Bob. The verb *to punish*, on the other hand, is an example of an NP2 verb because it biases towards the second noun phrase: The cause of punishing is assigned to the punished and not to the punisher. After reading *Bob punished Joe because he...*, the pronoun's antecedent is usually assumed to be Joe (Caramazza, Grober, Garvey, & Yates, 1977; Garvey, Caramazza, & Yates, 1974).

Another source of information that can be used to disambiguate the pronoun is the context the pronoun appears in. Combined with the reader's general knowledge, context information can cause one of the discourse entities to be seen as a more sensible referent. In *Bob lied to Joe because he could not handle the truth*, the context statement *he could not handle the truth* results in a preference for choosing *Joe* as antecedent: Not being able to handle the truth is known to be a reason to be lied to, more than a reason to lie. Context information does not agree with focus on the preferred referent in this example, so either of these sources of information may gain the upper hand during disambiguation. Alternatively, it is possible that the pronoun is not instantiated at all, or only partially (Greene, McKoon, & Ratcliff, 1992; Oakhill, Garnham, & Vonk, 1989).

In this section, standard DSS is extended with a pronoun resolution process resulting in a new model called DSS-PR that simulates how pronoun resolution is influenced by context information and focus arising from first mention and implicit causality. For adding the pronoun resolution process, no change to the standard model needs to be made whatsoever. Also, the same microworld is used as before, with the only difference that Jilly was turned into a boy named Joe in order make ambiguous pronouns possible.

5.2.1 The model

The standard DSS model bases its inferences on the situations described by a story text, without making use of any textual cues. For instance, the vector representation of the statement *he wins* (in a story about Bob and Joe) ignores the pronoun and any focusing, and equals $\mu(\text{B WINS} \vee \text{J WINS})$, meaning *Bob wins or Joe wins*. Contrary to this, in the pronoun resolution (PR) model that will be added to the standard model, focus affects the initial interpretation of *he wins* and the presence of a pronoun signals that one of the discourse entities needs to be chosen as its antecedent.

Focusing

When constructing the vector representation of a pronoun-containing statement, focusing is taken into account by letting the more focused discourse entity have a greater impact on the representation than the less focused entity. In a text about Bob and Joe in which it is stated that *he wins*, the corresponding situation is represented by a vector that is like $\mu(\text{B WINS} \vee \text{J WINS})$ but moved slightly into the direction of $\mu(\text{B WINS})$ if only Bob receives focus. Similarly, if only Joe receives focus, the vector $\mu(\text{B WINS} \vee \text{J WINS})$ is shifted towards $\mu(\text{J WINS})$. If both Bob and Joe are focused to some extent, both shifts are applied.

In general, assume that the description of the situation at time step t contains an ambiguous pronoun. Choosing Bob as the pronoun's referent would turn this situation into proposition p, while choosing Joe would result in proposition q. The initial vector representation of this situation then equals

$$X_t(0) = \mu(p \vee q) + \phi_p(\mu(p) - \mu(p \vee q)) + \phi_q(\mu(q) - \mu(p \vee q)), \tag{5.3}$$

where ϕ_p and ϕ_q are parameters that control the total focus received by Bob and Joe, respectively. Both focus parameters are positive and their sum cannot exceed 1. Equation 5.3 states that the vector representing the disjunction of p and q is moved towards $\mu(p)$ by the focus on Bob, while it is moved towards $\mu(q)$ by the focus on Joe.

The focus parameters ϕ_p and ϕ_q are composed of contributions from several sources of focus. Two such sources are considered here: the first-mention effect and implicit causality. In the simulations presented below, the first-mention effect contributes an amount of .2 and implicit causality an amount of .3 to the focus parameter. Assuming that Bob is mentioned first in the sentence being processed, this means that $\phi_p = .5$ and $\phi_q = 0$ if Bob is also focused by implicit causality (i.e., there is an NP1 verb), while $\phi_p = .2$ and $\phi_q = .3$ if implicit causality highlights Joe (i.e., in case of an NP2 verb).

Choosing an antecedent

If the statement *he wins* occurs in a text about Bob and Joe, it can be thought of as logically identical to *Bob wins or Joe wins*. During reading, however, there is an important difference between the two statements. Reading *Bob wins or Joe wins* might lead to all sorts of inferences, among which possibly one regarding who is most likely to be the winner. On the other hand, the statement *he wins* signals that the winner has to be determined by resolving the pronoun, resulting in either B WINS or J WINS. Therefore, pronoun resolution can be thought of as a form of inference aimed at choosing from a small set of possibilities.

In the PR model, this idea is implemented by defining *attractor regions* in situation space. Each attractor region corresponds to the result of choosing one of the possible antecedents of the pronoun. For example, the statement *he wins* in a text about Bob and Joe results in two attractor regions, one is the B WINS-region, the other the J WINS-region. The B WINS-region consists of all the points in situation space that represent situations in which Bob wins, while the J WINS-region contains all situations in which Joe wins.

Formally, the attractor region of a proposition p is defined by its vector representation $\mu(p)$. Given a point X_t, the vector $D_p = (d_{p,1}, \ldots, d_{p,150})$ such that

$d_{p,i} = \min\{0, \mu_i(p) - x_{i,t}\}$ points from X_t to the nearest point in the p-region. The distance from X_t to the p-region equals the euclidean length $||D_p||$ of this vector. The point X_t lies within the p-region if and only if this distance equals 0, that is, $||D_p|| = 0$.

The process of pronoun resolution is now quite simple. It begins by taking the vector representing the statement with the ambiguous pronoun and modified by focusing (i.e., vector $X_t(0)$ of Equation 5.3), and lets each attractor region 'pull' it into its direction, as explained in detail below. The influence an attractor region has on the vector increases as the vector gets closer to the region, making the vector 'fall' towards one of the regions with increasing velocity and resulting in an increased belief value for the corresponding proposition.[1] As soon as the vector arrives in one of the attractor regions, an antecedent is chosen and the pronoun is resolved. From then on, the attractor regions do not influence the vector any longer.

Assuming there are two attractor regions, one for p and one for q, this process is formalized as follows: If either $||D_p|| = 0$ or $||D_q|| = 0$, that is, X_t lies inside an attractor region, the pronoun is resolved and neither region affects X_t. Otherwise, the velocity of X_t in the direction of the p-region, denoted V_p, equals

$$V_p = \frac{D_p}{||D_p||(1 + ||D_p||)}. \tag{5.4}$$

That is, vector D_p is divided by its own length to result in the unit vector pointing from X_t towards the p-region. Next, it is divided by $1 + ||D_p||$ to result in a velocity that increases as the distance to the p-region decreases, until it reaches an arbitrary maximum of 1 when X_t is infinitesimally close to the p-region. The velocity of X_t caused by the two attractor regions together is its velocity resulting from pronoun resolution: $V_{\mathrm{PR},t} = V_p + V_q$.

Context influence

The pronoun resolution process described above is only influenced by focus and not by the information present in the context. The effect of context information can be modeled by simply adding standard DSS to the pronoun resolution model, resulting in the DSS-PR model. This addition is possible because

[1] A similar, gravity-inspired process forces a choice between possible word interpretations in the Visitation Set Gravitation model (Tabor, Juliano, & Tanenhaus, 1997; Tabor & Tanenhaus, 1999).

both models compute the movement of vectors (i.e., their velocity) through situation space. By adding the velocity according to standard DSS to the velocity according to PR, pronoun resolution becomes context dependent.

Let $V_{\text{DSS},t}$ denote the change in situation vector X_t over processing time according to standard DSS, as defined by Equation 4.9. The change in X_t caused by the combination of the standard DSS model and pronoun resolution, is now defined by the differential equation

$$\dot{X}_t = V_{\text{DSS},t} + \beta V_{\text{PR},t}, \tag{5.5}$$

that is, the change in vector X_t over processing time equals its velocity according to DSS plus β times its velocity according to PR. Parameter β controls the strength of the influence of the pronoun, and is set to a value of $\beta = .5$ in the simulations presented here.

The initial value for the evaluation of Equation 5.5 is $X_t(0)$ of Equation 5.3. If there is no pronoun but the text simply states that p is the case, the situation is of course not represented by the vector for the disjunction $p \vee q$ but by $\mu(p)$. This means that Equation 5.3 reduces to

$$X_t(0) = \mu(p) + \phi_q(\mu(q) - \mu(p)),$$

so $X_t(0) = \mu(p)$ for $\phi_q = 0$, as in the DSS model. That is, in the absence of a pronoun and of focusing, the DSS-PR model reduces to the DSS model, showing that the DSS-PR model is a true extension of the DSS model and that the results of the DSS model remain valid for DSS-PR.

At any moment during processing, belief values $\tau(p|X_t)$ and $\tau(q|X_t)$ can be observed to determine the extent to which the pronoun is instantiated to each of the entities. The process halts when, according to Equation 4.10, the criterium for sufficient processing is reached. This criterium is controlled by the depth-of-processing parameter θ, set to the value of $\theta = 0.3$ used in the standard model, unless stated otherwise.

As an example, take the sentence *Bob is tired and Joe is not, so he wins*, which describes two situations. The first is represented by the vector $\mu(\text{B TIRED} \wedge \neg(\text{J TIRED}))$ in situation space. Assuming that only Bob receives focus, the second situation, in which someone wins, is initially represented by a vector closer to $\mu(\text{B WINS})$ than to $\mu(\text{J WINS})$, and is attracted more by the B WINS-region than by the J WINS-region. However, the context statement *Bob is tired and Joe is not* makes it more likely that Joe is the intended referent of *he*. As a result, the vector is moved towards the J WINS-region by the standard model's inference process. Which of the entities is eventually chosen as antecedent depends on the informativeness of the context. If the context lends enough support to the inference that Joe is the winner, focus may be overruled to choose Joe as the pronoun's referent. It is also possible that the pronoun is only instantiated partially, meaning that the vector does not end up inside any attractor region. This is especially likely when processing is shallow (θ is small) and the effects of focusing and context are opposite.

Context strength

The influence that context is expected to have on the choice of the pronoun's antecedent can be expressed in terms of belief values. Without any context, the belief value of p is the a priori value $\tau(p)$. Assuming that t is the story time step at which the ambiguous pronoun occurs, the influence of the previous or following context situation $X_{t\pm1}$ changes the belief value by an amount $\tau(p_t|X_{t\pm1}) - \tau(p)$. Of course, the same is true for the belief value of q, the other possible result of resolving the pronoun. The context's preference for p over q equals the difference between the two context influences:

$$(\tau(p_t|X_{t\pm1}) - \tau(p)) - (\tau(q_t|X_{t\pm1}) - \tau(q))\,.$$

If this value is positive, context prefers p to result from choosing a referent. If it is negative, the context points towards q. The *context strength* is simply the absolute value of the context's preference. A context strength of 0 means that the context does not help at all to decide who the winner is. If, on the other hand, context strength is large, information from the context makes one of the characters much more likely to win than the other one. That is, context strength is a measure for the informativeness of the context for choosing the pronoun's referent.

5.2.2 Empirical findings

Before turning to the model's results, we present the empirical data against which the model will be validated. These fall into three categories. First, we shall look into the time course of pronoun resolution. Second, results regarding sentence reading times are discussed. Third, we shall discuss some findings concerned with errors in pronoun resolution.

The time course of pronoun resolution

At least two studies have indicated that an ambiguous pronoun can be (partially) instantiated based on focus, before disambiguating context information is available. Arnold, Eisenband, Brown-Schmidt, and Trueswell (2000) had subjects listen to short texts that brought one of two entities into focus. At the same time, these subjects looked at a picture corresponding to the situation described in the text. Eyetracking revealed that the subjects looked at the focused entity at the moment the text contained an ambiguous pronoun. It was not until somewhat later, when the pronoun could be disambiguated, that they looked at the pronoun's intended referent. It was concluded that focus can drive the initial instantiation of a pronoun and that the intended referent is chosen when context information becomes available. The same conclusion was drawn by Gordon and Scearce (1995) based on reading time data.

Reading times

One of the phenomena most studied in relation to pronoun resolution is implicit causality, which affects reading times. Congruent sentences are sentences in which the implicit cause agrees with context information, such as *Bob lied to Joe because he could not tell the truth.* These are generally read faster than incongruent sentences like *Bob lied to Joe because he could not handle the truth* (Garnham, Oakhill, & Cruttenden, 1992; Stewart, Pickering, & Sanford, 2000; Vonk, 1985).

In three self-paced reading experiments conducted by Stewart et al. (2000), subjects read congruent and incongruent sentences containing an ambiguous pronoun (the 'pronoun anaphor'-condition), and the same sentences in which the pronoun was replaced by the name of the intended antecedent (the 'name anaphor'-condition). To make sure that the subjects engaged in pronoun resolution, they were given questions that could only be answered if the pronoun was resolved correctly. The same questions were asked in the name-anaphor condition. It was found that there was a causal congruency effect in both conditions, without even an interaction between congruency and anaphor type. This shows that implicit causality is not restricted to sentences containing ambiguous pronouns.

In another experiment, Stewart et al. (2000) investigated the effect of processing depth on the congruency effect by varying the questions subjects had to answer. In the deep condition, the questions were the same as in the previous experiments. In the shallow condition, the pronoun did not need to be resolved to answer the question, supposedly resulting in shallower processing. It was found that the congruency effect was smaller in the shallow processing condition than in the deep processing condition.

Error rates
An ambiguous pronoun is not always instantiated to the referent that is most likely according to context information. When the context-inconsistent entity is chosen as referent, this is considered an error in pronoun resolution. Leonard, Waters, and Caplan (1997a) showed that the amount of context information affects the chance of making an error. They had subjects read sentences containing an ambiguous pronoun as well as disambiguating context information. The amount of context was varied by sometimes adding another full sentence that strongly implied one of two characters as the pronoun's referent. The subjects had to decide as quickly as possible to whom the pronoun referred. It was found that adding a context sentence resulted in fewer errors.

Congruency between implicit causality and context information not only influences reading times, but also has an effect on error rates. Leonard et al. (1997a, 1997b) and Stewart et al. (2000) found that more errors in pronoun resolution are made when reading incongruent sentences than when reading congruent sentences. Leonard et al. (1997a) also found that this effect becomes weaker when extra context information is added.

5.2.3 Simulation results

The DSS-PR model processed items containing the statement *he wins* at time step t in varying contexts. One example of a context is the story situation B TIRED $\wedge$ $\neg$(J TIRED) at $t-1$, which results in an item corresponding to the sentence *Bob is tired and Joe is not, so he wins*. As contexts, all situations were used that satisfied the following three constraints:

- the situation takes place at time step $t-1$ or $t+1$;
- it consists of a conjunction $p \wedge q$ of (negations of) basic propositions (where $p \neq \neg q$, but possibly $p = q$);
- it results in a context strength of at least .02.

This resulted in a total of 368 different contexts. Each processed item consisted of the context and the statement *he wins*, and was assumed to be embedded in a text about Bob and Joe, who served as the two possible antecedents of the pronoun.

Time course of pronoun resolution

The left panel of Figure 5.3 shows how the belief values of B WINS and of J WINS develop over processing time as the sentence *Bob is tired and Joe is not, so he wins* is processed with only Bob in focus. The right panel shows the development of belief values when the context is J TIRED $\wedge$ $\neg$(B TIRED) (*Joe is tired and Bob is not*) with only Joe receiving focus.[2] Clearly, the focused entity is preferred initially but this is overruled by context information preferring the other entity as antecedent.

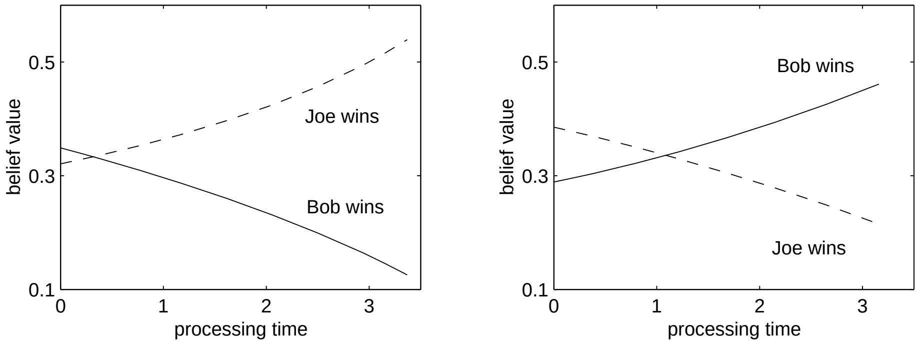

Figure 5.3: Belief values of B WINS and of J WINS during processing of *Bob is tired and Joe is not, so he wins* (left) and of *Joe is tired and Bob is not, so he wins* (right) with focus on the entity mentioned first.

Reading times

From here on, two separate sources of focus shall be modeled: one resulting from the first-mention effect, and the other from implicit causality. This means that each of the 368 items was processed four times: Either Bob or Joe is 'mentioned first' (i.e., receives focus because of first mention) and there can be an

[2] Compared to the left panel, the right panel shows a larger effect of focusing and a smaller effect of context, resulting in partial pronoun instantiation. This is partly caused by an asymmetry in the microworld: For no particular reason, Joe is a priori more likely to win than Bob, which is reflected in the belief values.

'NP1 verb' or an 'NP2 verb' (i.e., implicit causality adds focus to either the first mentioned or the second mentioned entity).

As discussed in Section 5.2.2, congruent sentences are read faster than incongruent ones. Stewart et al. (2000) found that this congruency effect on reading times occurred not only in sentences containing an ambiguous pronoun, but also in sentences where the pronoun was replaced by the name of the intended referent. There was no significant interaction between congruency and the type of anaphor (pronoun or name).

As Figure 5.4 shows, the model predicts processing times structurally similar to the reading times found by Stewart et al. Incongruent items were processed more slowly than congruent ones, and items containing the statement *he wins* were processed more slowly than corresponding items in which the intended winner was stated by proper name. Also, there is hardly any interaction between congruency and type of anaphor.

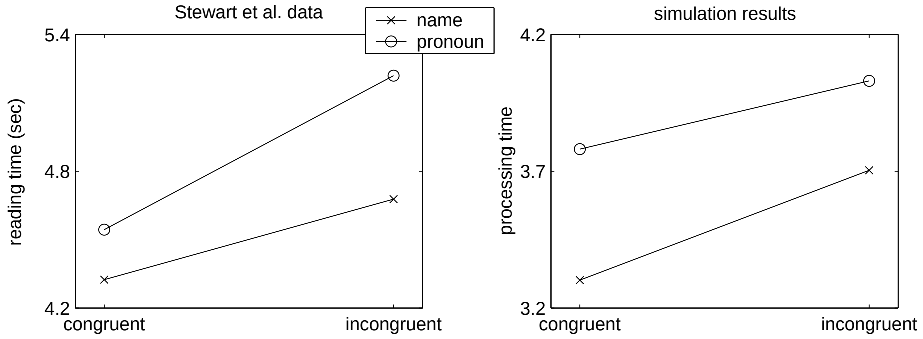

Figure 5.4: Left: Effect of congruency and anaphor type on reading times. Data from Stewart et al. (Table 2) averaged over implicit cause conditions NP1 and NP2. Right: Effect of congruency and anaphor type on model processing times, averaged over two directions of implicit-causality focus.

Another finding by Stewart et al. (2000) was that changing the subjects' reading task to one that did not require deep processing resulted in a smaller congruency effect. The model shows the same result (Figure 5.5). The congruency effect (defined as processing times on incongruent sentences minus processing times on congruent sentences) is smaller for values of the depth-of-processing parameter lower than $\theta = 0.3$, which was used in the previous simulations. Moreover, the model predicts an interaction between processing depth and anaphor type: The effect of θ is larger in the name-anaphor condition than in the pronoun-anaphor condition. Stewart et al. do not state whether the same interaction is significant in their data.

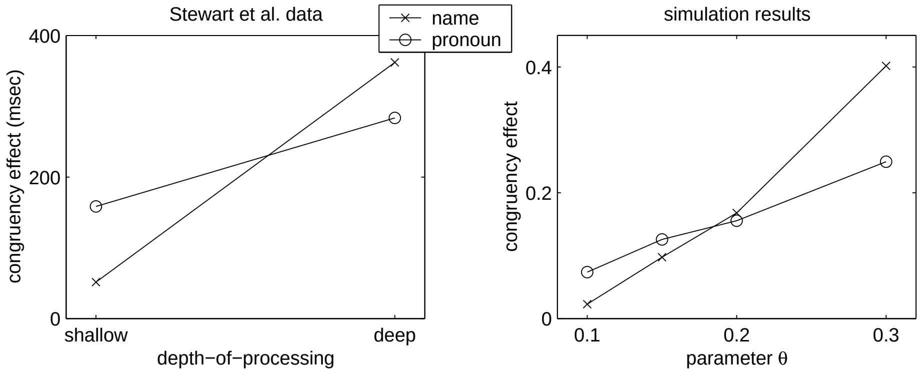

Figure 5.5: Left: Effect of processing depth on congruency effect. Data from Stewart et al. (Table 4, Fragment 2 reading times) averaged over implicit cause conditions NP1 and NP2. Right: Effect of depth-of-processing parameter θ on congruency effect in model simulations, averaged over two directions of implicit-causality focus.

All results for the figures above were averaged over the two implicit causality conditions. However, in all four of their experiments, Stewart et al. found that the congruency effect was strongly asymmetric in the pronoun condition: The effect was much smaller (sometimes even absent) for NP2 verbs, although not always significantly so. The same was found by Garnham et al. (1992, Experiments 4 and 5).[3] Stewart and Gosselin (2000) suggested that this asymmetry may be caused by the first-mention effect. An NP1 verb biases towards the subject, which is already in focus because it is mentioned first. This makes the preference for the subject even stronger, and results in a larger effect of congruency. An NP2 verb, on the other hand, biases towards the object, counteracting the first-mention effect. As a result, it does not matter much whether or not context information and implicit cause are congruent.

In the model, a processed item can be viewed as 'containing an NP1 verb' if the focus resulting from implicit causality highlights the same entity as the fo-

[3] Interestingly, in their first experiment, Garnham et al. (1992) found an unexplained larger congruency effect for NP2 verbs than for NP1 verbs.

cus caused by the first mention effect. If these two sources of focus are directed towards different entities, the item can be said to contain an NP2 verb. As Figure 5.6 shows, the model supports Stewart and Gosselin's hypothesis. The predicted congruency effect is quite similar to Stewart et al.'s findings. There is a larger congruency effect on reading times when first mention and implicit causality either both focus Bob or both focus Joe, compared to cases in which Bob is highlighted by one source of focus and Joe by the other.

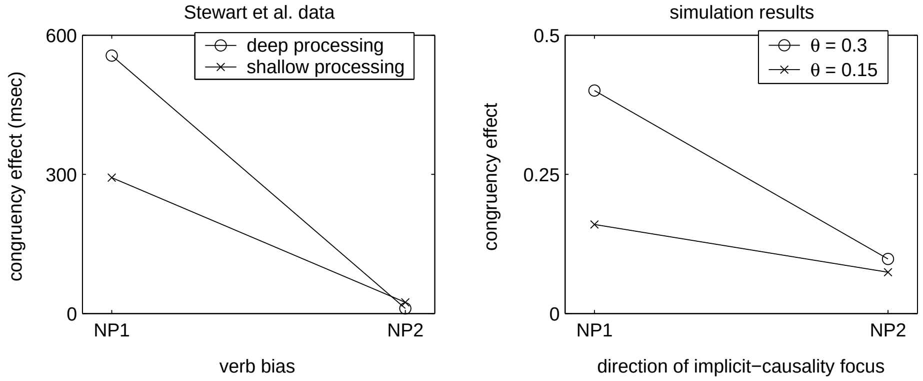

Figure 5.6: Left: Effect of verb bias on congruency effect. Data from Stewart et al. (Table 4, Fragment 2 reading times in pronoun condition). Right: Effect of focusing one or both entities on congruency effect in model simulations.

Error rates

An item was regarded as processed erroneously by the model when the final belief value of the context-inconsistent proposition was larger than that of the context-consistent proposition. Figure 5.7 shows that, although the model's error rate is too large when context is weak, it does predict the three effects found by Leonard et al. (1997a) and discussed in Section 5.2.2. First, incongruent items result in more errors than congruent ones. Second, stronger contexts result in lower error rates. Third, this latter effect was stronger for the incongruent items than for the congruent ones.

Stewart et al. (2000) found an asymmetric effect of implicit causality on error rates, similar to that found for reading times: The effect is larger for NP1 verbs than for NP2 verbs.[4] As Figure 5.8 shows, the model again predicts this asym-

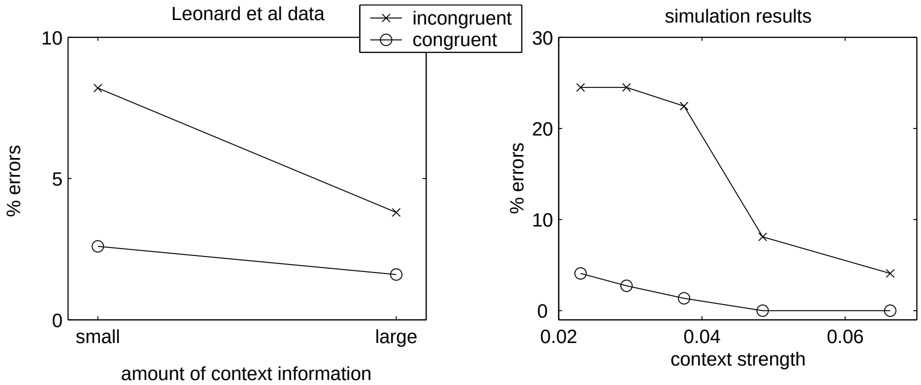

Figure 5.7: Left: Effect of congruency and context informativeness on error percentages. Data from Leonard et al. (1997a, Figure 6, 'young adults'-group). Right: Effect of congruency and context strengths on model error percentages, averaged over two directions of implicit-causality focus. Every point in the graph is the average of 147 or 148 processed items.

metry. The congruency effect on error rates is larger when one entity receives focus twice than in cases where one entity is focused by implicit causality and the other by first mention. Note that the predicted congruency effect on error rates is more extreme than found by Stewart et al. Nevertheless, the pattern in the data is predicted correctly. The absence of a congruency effect in the name condition is caused by the design of the standard DSS model, which does not allow for inferences that are inconsistent with text-given information. Only when focus is very strong and incongruent with context information can the incorrect entity be chosen. This happens just twice (out of 368 items) in the NP1 condition, and never in the NP2 condition.

5.2.4 Conclusion

The DSS-PR model successfully simulates the influence of context information and focusing on the resolution of ambiguous pronouns. The model arose as an extension to the standard DSS model by making two simple additions. First, focusing was assumed to have an effect on the initial representation of a text statement. Second, the choice between two possible interpretations of the pro-

[4] Stewart et al. do not present tests of the statistical significance of this interaction.

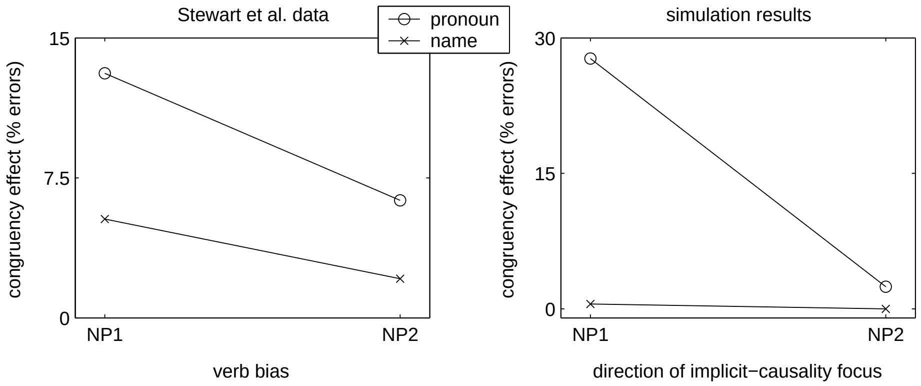

Figure 5.8: Left: Effect of verb bias on congruency effect. Data from Stewart et al. (Table 2). Right: Effect of focusing one or both entities on congruency effect in model simulations.

noun was forced by defining two attractor regions in situation space. This was sufficient to turn a model for inference into one for ambiguous pronoun resolution, which is in accordance with the assumption that ambiguous pronoun resolution is a special case of general knowledge-based inference.

The model is able to predict several empirical findings. First, it simulates how, over the course of processing time, context information overrules the initial, focus-based instantiation of a pronoun. This result is not surprising since focusing affects the initial vector representation of a pronoun-containing statement, and context begins to have an influence when this vector is adjusted by information present in the other statement.

Second, a large amount of reading time data was accounted for. As was found by Stewart et al. (2000), sentences containing ambiguous pronouns are processed more slowly than sentences containing names, and both these processing times are influenced by congruency between focus and context. Also, shallower processing leads to a smaller congruency effect, especially in the name-anaphor condition. Another result by Stewart et al. the model accounts for is the asymmetric effect of implicit causality. By assuming that both first mention and implicit causality affect focusing, it is explained why the congruency effect is larger for NP1 verbs than for NP2 verbs.

These results are of special importance because of the ongoing debate about the time course of implicit causality. Several studies have addressed the question at what point in the sentence implicit causality influences the comprehension process. According to the Focusing account, implicit causality arises while reading the verb, which influences discourse focus: An NP1 verb makes the subject more available to the reader, while an NP2 verb focuses attention on the object. Contrary to this, the Integration account claims that implicit causality does not affect comprehension until later, after the explicit cause was read, at the moment the two clauses of the sentence are integrated. There is disagreement among researchers concerning which account is correct: McDonald and MacWhinney (1995) find support for a focusing effect of implicit causality, while Garnham, Traxler, Oakhill, and Gernsbacher (1996) conclude that its effect takes place during integration. Long and De Ley (2000) claim that it depends on reading skill: Skilled readers show an earlier effect of implicit causality than less skilled readers do.

The results of these three studies were based on probe recognition tasks, which Stewart et al. (2000) criticize for interfering with the normal comprehension process and for not being sensitive enough to the time course of processing. They argue for taking reading time measures instead and claim the two accounts make different predictions about the effect of causal congruency on reading times. First, only the Integration account is claimed to predict a congruency effect in sentences where the ambiguous pronoun is replaced by the name of the intended antecedent. That is, the congruent sentence *Bob lied to Joe because Bob could not tell the truth* would be read faster than the incongruent *Bob lied to Joe because Joe could not handle the truth* according to the Integration account, but not according to the Focusing account. Second, only the Integration account would predict a smaller congruency effect when readers engage in shallower processing. Indeed, Stewart et al. did find a congruency effect in sentences without ambiguous pronouns and a smaller effect when processing was shallower, from which they concluded that the Focusing account is incorrect.

In the model, implicit causality was incorporated by making one of the entities more preferred before any integration of clauses takes place. That is, the model's results were based on a focusing account of implicit casuality. Nevertheless, the model predicts a congruency effect in the name-anaphor condition and a smaller effect as depth-of-processing parameter θ is decreased. This clearly shows that Stewart et al.'s conclusion was not warranted. Of course, the model does not claim that implicit causality must have its effect on focusing. It only shows is that such an account is consistent with empirical findings.

It is clear why there should be a congruency effect in both the name-anaphor and pronoun-anaphor conditions. Focusing leads to an expectation, and it takes some time to neutralize this expectation when new information is inconsistent with it, regardless whether a pronoun is used or a name. It is less obvious why the congruency effect is lower with shallow processing than with deep processing, since the effect of focusing does not depend on the level of depth-of-processing parameter θ. Part of the decrease in congruency effect with shallower processing is caused by a general decrease in processing time, but especially in the name-anaphor condition this is not enough to explain all of the effect. When the value of θ is low, an item is easily considered sufficiently processed. In other words, there is not much of a need to strongly integrate the two clauses of the sentence, making context information less relevant. Since congruency concerns the relation between focusing and context information, this means that congruency itself is less relevant when processing is shallow than when it is deep.

A third set of model results showed that the percentage of errors in pronoun resolution decreases as context strength increases, which is consistent with data by Leonard et al. (1997a). Strong contexts can overrule an inconsistent focus more often than weaker contexts can, resulting in fewer errors. Moreover, implementing implicit causality as a form of focusing not only accounts for the causal congruency effect and its asymmetry in reading times, but also in error rates. This lends more support to the hypothesis that the causal congruency effect results from focusing.

5.3 Towards a textbase-level representation

> A paper based on this chapter is submitted for publication as Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003c), *Modeling multiple levels of text representation*.

As far as the standard DSS model is concerned, a story is no more than a temporal sequence of situations. This makes story comprehension no different from understanding events going on in the real world. The reader of a text, however, can make use of textual information that is not available to an observer of real world events. In particular, causal connectives like *because* and *although* can influence the processing of a text (Millis & Just, 1994) and its recall (Millis, Graesser, & Haberlandt, 1993). The same is true for temporal connectives. For instance, the second of the three stories the standard model was tested on (see Section 4.5.2) states that Bob and Jilly are inside at $t = 3$. From this, it is inferred that they were also inside at $t = 2$, when playing hide-and-seek. It is not possible to tell the model that a new episode had started at $t = 3$ by adding a connective like *next* or *then*. Bestgen and Vonk (1995) showed that temporal markers like *then* reduce the availability of information in the previous sentence, so in the model such a connective could signal that the influence between the story situations at $t = 2$ and $t = 3$ should be decreased.

The previous section on pronoun resolution formed a first demonstration of the influence of textual cues on the model's inference process. Although it showed how focus and pronouns can affect the story's situational representation, Section 5.2 was not concerned with the representation of such textual information. This information is not present at the DSS model's situational level of representation. Instead, of the three levels of discourse representation discussed in Section 1.1.2, it is at the textbase level that textual cues reside. This raises the question whether such a textbase level can be added to the standard DSS model. Textual information carried by, for instance, connectives and pronouns is present at this textbase level and can influence the inference processes that takes place at the situational level.

In this section, a first step is made towards extending the DSS model with a more textual level of representation. This shall be accomplished by training a recurrent neural network to take as input sentences (i.e., word sequences) describing microworld situations and to transform them into the DSS-vector representations of these situations. This is quite similar to the task performed by St. John and McClelland's (1990, 1992) Sentence Gestalt model discussed in Section 2.6. One important difference between that model and ours is that the Sentence Gestalt model does not give a complete representation of the sentence's meaning as output, but only answers questions about the contents of the sentence. Also, the output representation of the Sentence Gestalt model is localist while ours is distributed.

A model developed by Desai (2002) to simulate language learning by children also consists of a recurrent network that transforms sentences into localist representations of their meaning. Contrary to the Sentence Gestalt model, these output representations do contain all the information in the sentence. A model even more similar to ours is the Connectionist Sentence Comprehension and Production (CSCP) model by Rohde (2002). It consists of a neural network that, like ours, learns to transform sentences into independently developed, distributed output representations. However, unlike our DSS vectors, the distributed output vectors of the CSCP model were not designed to represent statements at a situational level but only to encode and decode propositional structures. The relations between those vectors do not reflect probabilistic relations between the world events they represent.

The most important respect in which all three of the models mentioned above differ from the one presented in this section is that we are mainly concerned with the network's internal representation that develops during training. This internal representation, we shall argue, provides a radically different view of the traditional surface/textbase/situation-distinction in levels of text representation.

5.3.1 The microlanguage

The sentences the network learns to process are composed of 15 different words, most of which are also words in English: *Bob, Jilly, and, play, are, win, lose, soccer, hide-and-seek, a_computer_game, with_the_dog, outside, inside, tired, awake*. To simplify the already simple language, both *a_computer_game* and *with_the_dog* are considered one word. For further simplification, verbs are not inflected. Note that the microlanguage vocabulary lacks the word *not* and other negations. Kaup and Zwaan (2003) argue that processing a negation involves first constructing the situation model of the corresponding non-negated statement, and then directing attention away from it. Such a two-step process is beyond the network's capabilities.

The 15 words can be combined into sentences following the grammar of Table 5.1. In total, the microlanguage consists of 328 different sentences. Thirty-eight of these, shown in Table 5.2, are put aside as a test set. Since the network is not trained on these, it is not shown any sentences in which

- hide-and-seek is played outside (Group 1);
- anyone plays with the dog inside (Group 2);
- *Bob and Jilly* (in this order) play soccer (Group 3);
- *Jilly and Bob* (in this order) play a computer game (Group 4).

Moreover, some conjunctions only appear in one of the two possible orders (Group 5). For instance, the network is trained on *Bob play soccer and are tired*, but not on *Bob are tired and play soccer*. Note that the first two groups of test sentences describe situations not mentioned by any of the training sentences, while the last three groups consist of alternative descriptions of situations also present in the training set.

Table 5.1: Grammar of Bob and Jilly's microlanguage.

S	→	NP VP
NP	→	*Bob* | *Jilly* | *Bob and Jilly* | *Jilly and Bob*
VP	→	*play* Game [Place | *and are* State | *and* Result]
	→	*are* Place [*and play* Game | *and* State | *and* Result]
	→	*are* State [*and play* Game | Place | *and* Result]
	→	Result [*and play* Game | Place | *and are* State]
Game	→	*soccer* | *hide-and-seek* | *a_computer_game* | *with_the_dog*
Place	→	*outside* | *inside*
State	→	*tired* | *awake*
Result	→	*win* | *lose*

5.3.2 Training the network

Figure 5.9 shows the architecture of the recurrent neural network that learns to transform microlanguage sentences into the corresponding microworld situation vectors. The words of a sentence enter the network one by one. Each word

Table 5.2: Thirty-eight sentences used as a test set.

group	sentence
1	Bob play hide-and-seek outside
	Bob are outside and play hide-and-seek
	Jilly play hide-and-seek outside
	Jilly are outside and play hide-and-seek
	Bob and Jilly play hide-and-seek outside
	Bob and Jilly are outside and play hide-and-seek
	Jilly and Bob play hide-and-seek outside
	Jilly and Bob are outside and play hide-and-seek
2	Bob play with_the_dog inside
	Bob are inside and play with_the_dog
	Jilly play with_the_dog inside
	Jilly are inside and play with_the_dog
	Bob and Jilly play with_the_dog inside
	Bob and Jilly are inside and play with_the_dog
	Jilly and Bob play with_the_dog inside
	Jilly and Bob are inside and play with_the_dog
3	Bob and Jilly play soccer
	Bob and Jilly play soccer outside
	Bob and Jilly play soccer inside
	Bob and Jilly play soccer and are tired
	Bob and Jilly play soccer and are awake
	Bob and Jilly play soccer and win
	Bob and Jilly play soccer and lose
4	Jilly and Bob play a_computer_game
	Jilly and Bob play a_computer_game outside
	Jilly and Bob play a_computer_game inside
	Jilly and Bob play a_computer_game and are tired
	Jilly and Bob play a_computer_game and are awake
	Jilly and Bob play a_computer_game and win
	Jilly and Bob play a_computer_game and lose
5	Bob are tired and play soccer
	Bob are outside and tired
	Bob play hide-and-seek and are awake
	Bob are awake and win
	Jilly play a_computer_game and are tired
	Jilly are tired and inside
	Jilly are awake and play with_the_dog
	Jilly lose and are awake

is represented locally, by activating one of 15 input units. This activation is fed to the hidden layer, consisting of six units, which also receives its own previous activation state. As a result, the pattern of activation over the six units of the hidden layer forms a representation of the sentence read so far. When the last word of the sentence is processed, the activation pattern over the 150 output units should be the 150-dimensional DSS vector representing the situation described by the sentence, which can be used as input to the DSS model. The activation pattern of the hidden layer after processing a complete sentence shall be called the *intermediate representation* of the sentence, because it lies between the network's word-level input and its situation-level output.

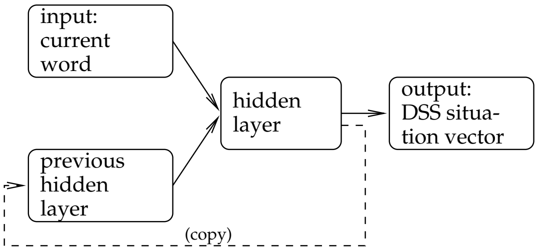

Figure 5.9: Architecture of the network used to develop intermediate representations of microlanguage sentences. The input layer has 15 units, one for each word. The hidden layer, consisting of six units, receives activation from the input layer and a copy of its own previous state. The output layer has 150 units, one for each situation-space dimension. A solid arrow between two layers indicates that the first layer is fully connected to the second. The dashed arrow indicates that the activations of the hidden layer are copied to the previous hidden layer.

Of course, the network has to be trained to produce the correct situation vector for each input sentence. During training, the output activations are compared to the correct situation vector whenever the complete sentence has been processed. For example, the network is shown the word sequence *Bob play soccer* and produces an output, which is compared to the situation vector μ(SOCCER).[5] Next, the error in the output is backpropagated to update the connection weights. The set of 290 training sentences was presented to the network 220 times, each time in a different random order. The network is trained

[5] The network's output to the sentence *Bob play soccer* cannot be compared to the situation vector μ(B SOCCER), because the basic proposition B SOCCER does not exist. The sentence describes the situation in which both Bob and Jilly play soccer, because they always play this game *together*.

seven times, with different random initial weight settings on each occasion. All results presented below are averaged over the results for these seven training sessions.

5.3.3 Amount of comprehension

To investigate whether the network learned to produce the correct situation vector for each training input, the produced and correct output vectors are compared. This could be done by computing the mean squared error, but a more easily interpreted measure is available by using belief values. Assume the input sentence describes proposition p and the network's output is the vector X_p. If the network has not learned anything, we may expect $\tau(p|X_p)$, the belief value of p in the situation represented by X_p, to equal the a priori belief value $\tau(p)$. In that case, the network's 'amount of comprehension' of the sentence is 0. If $\tau(p|X_p)$ is larger than $\tau(p)$, the sentence can be said to be 'understood' to some extent. In the ideal case, when $X_p = p$ so $\tau(p|X_p) = \tau(p|p)$, the amount of comprehension is defined to equal 1. If, on the other hand, $\tau(p|X_p)$ is smaller than $\tau(p)$, the sentence is misunderstood and the amount of comprehension is negative. Formally, the amount of comprehension of the sentence by the network equals

$$\text{compr}(p) = \frac{\tau(p|X_p) - \tau(p)}{\tau(p|p) - \tau(p)}. \tag{5.6}$$

Most microlanguage sentences form a conjunction of two statements. In that case, the comprehension measure of Equation 5.6 can be somewhat misleading. For instance, if the sentence *Jilly play hide-and-seek outside* results in an output vector that is identical to μ(J OUTSIDE), the network has not understood that Jilly plays hide-and-seek but only that she is outside. Nevertheless, the amount of comprehension will be positive because the belief value of HIDE-AND-SEEK $\wedge$ J OUTSIDE is larger given J OUTSIDE than a priori. Therefore, for sentences describing a conjunction $p \wedge q$, the amount of comprehension of a conjunction is also computed for p and q separately. Note that the amount of comprehension for the individual statements is computed after processing the *complete* sentence, that is, the conjunction $p \wedge q$.

5.3.4 Results

Learning and generalization

Table 5.3 shows the average amounts of comprehension for the sentences in the training and test sets, for the complete sentence as well as for the first and second statements separately. All values are significantly positive, indicating that the network does learn to comprehend the training sentences above chance level and generalizes this skill to test sentences. However, first statement comprehension is quite poor, especially for the test sentences. The second statement often seems to override the information in the first.

Table 5.3: Amounts of comprehension, averaged over n values, and 95% confidence interval for training and test sentences, both for the complete statement and separately for the first and second statement of a sentence describing a conjunction.

			statement	
set	n	complete	first	second
training	2030	.28 ± .01	.18 ± .02	.56 ± .01
test	266	.20 ± .03	.06 ± .04	.62 ± .03

It is also informative to look at the percentages of misunderstood sentences (i.e., resulting in a negative amount of comprehension). The error rates closely follow the amounts of comprehension. Again, first statements are often processed poorly: The error percentages for training sentences are 25.2% and 0.8% for the first and second statement respectively. For test sentences, almost half of the first statements are misunderstood, as can be seen from Table 5.4. However, these errors are not divided evenly over the 38 test sentences. The network seems to have particular difficulty learning to process sentences that describe new situations (Groups 1 and 2). The first statement of such sentences seems to be completely overwritten by the second. In comparison, the network had more success learning that the connective *and* is commutative (Groups 3 to 5), so novel descriptions of previously trained situations are processed reasonably well.

The surprisingly large first statement error rates and negative comprehension scores for Group 2 test sentences can be explained by the microworld situations these sentences refer to. They are all about playing with the dog inside, but Bob and Jilly are more likely to play with their dog *outside*: The a priori belief value of Bob and Jilly being inside is $\tau(\neg\text{B OUTSIDE} \wedge \neg\text{J OUTSIDE}) = .26$,

Table 5.4: Error percentages and amounts of comprehension, averaged over n values, for test sentences, both for the first and second statement of a sentence describing a conjunction, per test sentence group and averaged over all test sentences. Group numbers refer to Table 5.2.

		statement			
		first		second	
group	n	% err	compr	% err	compr
1	56	64.3	$-.15$	0.0	.78
2	56	75.0	$-.05$	0.0	.68
3	49	31.0	.31	0.0	.50
4	49	16.7	.16	2.4	.44
5	56	35.7	.10	1.8	.62
av.	266	46.8	.06	0.8	.62

while the belief value given that they play with the dog, is only $\tau(\neg\text{B OUTSIDE} \wedge \neg\text{J OUTSIDE}|\text{B DOG} \wedge \text{J DOG}) = .12$. This means that understanding only half of a sentence in which Bob and Jilly play with the dog outside will reduce the belief value (and thereby the amount of comprehension) of the other half. Similarly, the large first statement error rates for test sentences in Group 1 are caused by the fact that hide-and-seek is more likely to be played inside, contrary to what these sentences state. Given that Bob and Jilly play hide-and-seek, the belief value of them being inside increases to $\tau(\neg\text{B OUTSIDE} \wedge \neg\text{J OUTSIDE}|\text{HIDE-AND-SEEK}) = .36$

The intermediate representation

Recall from Section 1.1.2 the experiment by Fletcher and Chrysler (1990) from which they concluded that there exist three distinct levels of discourse representation: the surface text, the textbase, and the situation model. In this experiment, subjects more often confused two sentences that differed only at the surface-text level than two sentences that differed also at the textbase level. A similar distinction can be made with sentences in our microlanguage. The sentences *Bob and Jilly play soccer* and *Jilly and Bob play soccer* differ at the surface level but, supposedly, not at the propositional level since the commutative property of AND makes AND(BOB,JILLY) the same proposition as AND(JILLY,BOB). Contrary to this, the sentences *Bob play soccer* and *Jilly play soccer* differ both as surface texts and as propositions. They describe the same situation, however, because soccer is always played by both Bob and Jilly.

Eight pairs of sentences about *Bob and Jilly* and their *Jilly and Bob* counter-parts form the so-called 'surface different' set of sentence pairs, shown in Table A3 of Appendix A.2. The two sentences of each of these pairs differ only at the surface text level. The 10 'textbase different' sentence pairs, also shown in Table A3, describe different propositions but identical situations.

Fletcher and Chrysler's subjects were also more likely to confuse two sentences that differed only at the surface text and textbase levels than two sentences that differed at the situational level as well. Again, this distinction can be made in our microlanguage. The sentence pair *Bob play soccer* and *Jilly play soccer*, like all other pairs in the 'textbase different' set, differ propositionally but not situationally. The pair *Bob play with_the_dog* and *Jilly play with_the_dog*, on the other hand, differ at both the textbase and the situational level. Ten of such sentence pairs, given in Table A3 of Appendix A.2, form the 'situation different' set.

Directly modeling Fletcher and Chrysler's experiment would require the implementation of some kind of word recognition process. We propose that this difficulty can be circumvented by taking the sentences' intermediate vector representations and assuming that similar vectors are more difficult to tell apart than dissimilar ones. This implies that similarity in the intermediate representations corresponds to confusability of the sentences.

As a measure of dissimilarity of two vectors, the euclidean distance between them is used. For each of the seven trained networks, the distances between the 328 vectors for all microlanguage sentences are normalized to an average of 1. Figure 5.10 shows the normalized distances between the vector representations of sentence pairs from the three different sets, averaged over seven repetitions of 8 distances for the 'textbase different' set and of 10 distances for the other two sets. The distances follow the percentages of correct responses found by Fletcher and Chrysler: Sentences that differ only in surface text are more similar to one another than sentences that differ also propositionally but not situationally ($t_{124} = 9.38; p < .001$), which in turn are more similar to one another than sentences that do differ situationally ($t_{138} = 2.05; p < .05$).

5.3.5 Conclusion

Although the network's performance was far from impressive, it did learn to comprehend both training and test sentences significantly above chance level, with the exception of the first statement of sentences describing situations not

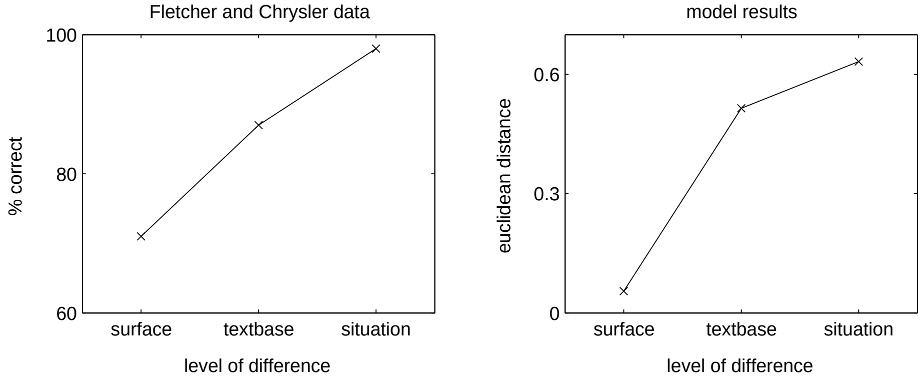

Figure 5.10: Left: experimental results by Fletcher and Chrysler (1990). Right: distances between vectors representations in the network's hidden layer, for sentences that differ only at the surface level, sentences that differ only at the surface and textbase levels, and sentences that differ also at the situational level.

mentioned in the training set (test sentence Groups 1 and 2). No matter how important it is to make sure that the network did learn its task, this result is only of secondary interest since we are mainly concerned with the intermediate representation that developed during training.

The intermediate vector representation is neither fully based on surface text nor purely situational. Any difference between two sentences increases the distance between the corresponding vectors, regardless whether the difference is one of surface text, proposition, or situation. Two sentences that describe different situations necessarily also form different propositions, so they will be at least as different from each other as two sentences that differ only propositionally. Likewise, two sentences that differ propositionally must also differ in surface form, so they will be at least as different from each other as two sentences that differ only in surface form. Since any difference between sentences adds to the distance between the corresponding intermediate vector representations, vectors are less similar to one another if the sentences they represent differ at a higher level. This indicates that a *single* representation can encode information about the surface text, the proposition, and the situation. Interestingly, such a representation can predict Fletcher and Chrysler's (1990) findings even though they took their results as providing "strong converging evidence for the psychological reality of van Dijk and Kintsch's (1983) distinction among surface memory, the propositional textbase, and the situation model" (p. 177).

We do not claim that only one level of representation exists. In fact, a purely situational representation was needed to train the network and develop the intermediate representation. It is clear, however, that Fletcher and Chrysler's result does not require three levels of representation. One property of distributed representations is that they can simultaneously encode different aspects of the represented item. Therefore, one sentence vector can represent, to some extent, surface text, proposition, and situation.

Traditionally, constructing a textbase is viewed as one of the main goals of text comprehension. Such a textbase is assumed to consists of propositional structures. Our simple network challenges both these views. First, textbase-like properties may be part of an intermediate representation that only exists because it is necessary, or at least useful, for the construction of the situation model. Second, there is no need for propositional structures. The vector that represents a sentence at the intermediate level cannot be decomposed into a predicate and its arguments.

Of course, all of these conclusions are tentative. The network only processes single sentences, while a textbase should be able to include several sentences. Also, our microworld and microlanguage are extremely simple, so research is needed with worlds and languages of more realistic size. It is in fact not unlikely that this will increase the textbase-like character of the intermediate representations, since it may be the need to comprehend complex language that drives the development of representations that, in some sense, can be regarded propositional.

The motivation for extending the standard model with a sentence comprehension network was the development of a more textual representation of story situations. The intermediate representation that resulted is indeed more text-like than the DSS situation vectors. However, the possibility of incorporating textual cues into the representation has not yet been investigated. It is in fact still unclear how such a thing can be accomplished. Of course, the microlanguage could be extended with connectives and pronouns but this would not change the situations to which sentences refer. Therefore, training the network to produce situation vectors from sentences of such an extended microlanguage will not result in the textual information being used as processing cues.

6
Summary and conclusions

6.1 Summary

Comprehending a discourse takes more than comprehending its individual sentences because sentences in a discourse are related to one another. Without taking these relations into account, a discourse cannot be fully understood. This thesis deals with theories of the discourse comprehension process that are specified in enough detail to be implemented and executed as computer programs. Such theories are called computational models.

6.1.1 Eight models of discourse comprehension

Seven computational models of discourse comprehension were discussed in Chapter 2: the Resonance model, the Landscape model, Langston and Trabasso's model, the Construction-Integration model, the Predication model, the Sentence Gestalt model, and the Story Gestalt model. Chapter 3 added the Golden and Rumelhart model to this list.

The Resonance model (Myers & O'Brien, 1998) simulates how incoming sentences activate the mental representation of concepts and propositions from the preceding text. It shows how text items disappear from working memory when they are not related to the sentence being read, and how they can be reinstated by reading a word they appeared with before. Although the model simulates this process successfully, it does suffer from several problems. First, it is likely that its parameter setting is not text-independent. Second, some values of variables increase to unrealistically high levels during the model's process. Third, it does not incorporate the reader's world knowledge even though this is sometimes crucial for finding connections within a text.

Activation of text items in working memory leads to an episodic memory trace of the text. The simplest of the seven models from Chapter 2, the Landscape model (Van den Broek, Risden, Fletcher, & Thurlow, 1996), takes as input hypothetical activations of concepts in working memory, and computes from these activations the strengths of items, and of the relations between them, in a text's memory trace. More specifically, the model is an implementation of the generally accepted ideas that text items obtain a stronger memory trace if they occur in working memory more often, and that two text items become more strongly associated to each other in the text's memory trace the more often they occur together in working memory. The Landscape model's strength lies in its simplicity, which at the same time is its main weakness. Because the model's output results so directly from its input, it does not produce any surprising outcomes or explain much about the process of concept activation and memory trace formation.

The model by Langston and Trabasso (1999) is the first of the models we discussed that incorporates the reader's knowledge of the world. It focuses on the causal relations between events, taking as input a list of possible causal connections between the sentences being processed. It is then claimed to compute which of these connections would actually be made by a reader. However, it was shown that this model does not predict anything but the undesired effect that early sentences receive stronger causal connections. Moreover, it requires all possible causal connections within a text to be given in advance, which is far from realistic.

The problem for any model that includes world knowledge is the huge amount of knowledge readers can apply when comprehending a text. No realistic part of all this knowledge can be implemented. By far the best known and most influential model of discourse comprehension, Kintsch's (1988, 1998) Construction-Integration model, handles this problem by assuming that comprehending a discourse statement proceeds in two phases. First, during the construction phase, items from the statement are associated with a few items from the reader's knowledge. The more related a text item is to a knowledge item, the more likely it is for the knowledge item to be selected. Only these associated items need to be implemented in the model. Statements from the discourse context do not influence the construction phase in any way. Instead, during the second phase, called integration, a context-sensitive process discards irrelevant or inconsistent items from the collection of text and knowledge items. What remains is the reader's interpretation of the text.

Although the Construction-Integration model can be praised for recognizing the importance of knowledge to discourse comprehension, it does not live up to its reputation. Apart from some serious technical difficulties with the integration process, we have argued that its basic idea does not suffice. A context-independent construction process cannot generally come up with the required associations from the reader's knowledge, unless it produces so many associations that it loses any explanatory power.

The next model we discussed was the Predication model (Kintsch, 2001). This model is an extension to Latent Semantic Analysis (Landauer & Dumais, 1997), which is a method for developing vector representations of words. Similarities among vectors reflect semantic similarities among the words they represent. In this way, the vectors encode part of the words' meaning. The Predication model aims at explaining how representations for propositions can be constructed from these word vectors. However, propositions are related to truth values and, so we argued, these can never arise from simply combining the meaning of individual words. Therefore, the Predication model can only adjust word vectors to context, but not produce representations of propositions.

The Sentence Gestalt model (St. John & McClelland, 1990, 1992) does develop vector representations of propositions. The model is basically a recurrent neural network that is trained to process sentences (i.e., word sequences) and answer questions about the events they describe. During this training, it develops vector representations of the sentences. The architecturally identical Story Gestalt model (St. John, 1992) learns to process stories (i.e., sequences of propositions) and to answer questions about a story's events. In order to perform this task, it needs to develop vector representations of the stories.

Once trained, the Sentence Gestalt model is able to fill in inferable information missing from sentences. Likewise, the Story Gestalt model uses its acquired knowledge of stories to draw inferences about events that must have occurred but are not stated in the story being processed. For instance, it can find the most likely referent of a pronoun. The main limitations of the two Gestalt models are that they cannot predict reading times and that the representations they develop are only useful for the task the networks were trained to perform. For the Story Gestalt model, this means that it knows nothing about the order of events in the story.

The model proposed by Golden and Rumelhart (1993) introduces the idea of a situation space. Vectors in this space represent situations that can occur in a story. Each dimension of the situation space corresponds to one proposition, and story situations consist of conjunctions of these propositions. A story, being a temporal sequence of situations, is represented as a trajectory through situation space. The model takes as input such a trajectory and transforms it into a more informative one, using knowledge about the causal relations between the propositions that make up the dimensions of the situation space. The model's mathematical foundations are formed by Markov random field theory, which guarantees that the trajectory resulting from the model is always the most likely given the constraints posed by the story's statements.

The Golden and Rumelhart model was shown to suffer from some limitations regarding the stories and causal knowledge it can implement. First, it cannot implement knowledge regarding constraints within a story situation. Second, only situations consisting of a single proposition or a conjunction of propositions can be represented. Third, the model cannot implement knowledge about a conjunction's cause or consequence if these are qualitatively different from those of the individual propositions that make up the conjunction, as is the case in the XOR relation. We have argued that all of these limitations result from some of the model's basic architectural assumptions, in particular the assumption that there is no influence among beliefs in propositions within one moment in the story. However, this assumption cannot be discarded without making the Markov random field analysis infeasible.

Analyzing these eight models revealed that computational soundness is often lacking in discourse comprehension models. Without mathematical rigor, so-called computational models are nothing more than verbal models disguised in equations. Only the Gestalt models and Golden and Rumelhart's model can be said to be computationally sound. Moreover, considering four of the model evaluation criteria proposed by Jacobs and Grainger (1994), progress in the field of discourse comprehension models looks bleak. To begin with, serious validation against empirical data is often lacking. Models that do seem to predict much data can often not be credited for this (e.g., the Langston and Trabasso model) or only do so because of ad hoc and subjective parameter settings (e.g., the Construction-Integration model). Even models that are validated against data often do not explain much (e.g., the Landscape model) or even nothing at all (e.g., Langston and Trabasso's model). Other models require many ad hoc equations and parameters (e.g., the Construction-Integration model), while some cannot readily be applied to several inputs (e.g., the Golden and Rumelhart model) or tasks (e.g., the Gestalt models).

6.1.2 The Distributed Situation Space model

In Chapter 4, an attempt was made to develop a model of knowledge-based inferencing during story comprehension that is not only computationally sound, but also predicts and explains empirical data, requires only few assumptions, can process different stories, and simulate several tasks. The resulting Distributed Situation Space (DSS) model shares most of its architecture with the Golden and Rumelhart model, guaranteeing computational soundness. The most important difference with the Golden and Rumelhart model lies in the representation of story situations. The DSS model's situation space has no one-to-one mapping between dimensions and propositions. Instead, the representation of a proposition is distributed over multiple dimensions, and each dimension codes for several propositions.

The situation space is developed by extracting information from a handcrafted description of events going on in a microworld. In this microworld there exist two characters who can be in different states and engage in several activities. A Self-Organizing Map (Kohonen, 1995) extracts regularities within microworld situations, and learns to represent microworld propositions accordingly. The advantages of this representation are threefold. First, any boolean combination of propositions (i.e., any situation in the microworld) can be represented as a vector in situation space. Situations are therefore not restricted to conjunctions of propositions. Second, the a priori subjective probability that a proposition or situation occurs in the microworld can be computed from the situation's vector representation. Third, conditional subjective probabilities given certain propositions or situations follow from the vector representations. The subjective probability of a proposition is called its belief value, because it indicates the extent to which a reader might believe the proposition to be the case in a story. Belief values are also used to define measures of proposition fit and story coherence, which are useful for validating the model's results against empirical data.

Like the Golden and Rumelhart model, the DSS model represents a story as a temporal sequence of situations, forming a trajectory through situation space. Knowledge about temporal relations between consecutive situations in the microworld is used to transform such a story trajectory into a more informative one, taking the constraints given by the story text into account. From the model's mathematical basis, Markov random field theory, it follows precisely how world knowledge regarding temporal relations between situations can be implemented, and how a story trajectory can be brought into closer correspondence to temporal world knowledge during the model's inference process. Meanwhile, belief values of propositions can be obtained to ascertain the extent to which these propositions are inferred. The point at which the inference process stops depends on a parameter that controls processing depth.

Results showed that propositions are indeed inferred if they are likely to be true in the story, given the regularities in the microworld. Moreover, the model predicts a fair amount of empirical data. In accordance with the data, it simulates how processing a story situation that is less related to the previous statement results in more inferences and longer processing times. Increased amounts of inference and processing time also result from increasing the value of the depth-of-processing parameter. This, too, has been found empirically.

6.1.3 Extending the DSS model

Three extensions to the DSS model were presented in Chapter 5. First, it was shown how the model easily simulates story retention. The trajectory resulting from the inference process can be viewed as a story's memory trace. Over retention time, this trace weakens, meaning that the amount of information in it decreases. By assuming that parts of the trajectory more strongly related to each other lose their information more slowly, the retention model simulates how propositions that contribute least to the story's coherence are the first to be forgotten. Moreover, it correctly predicts that less is recalled as retention time grows, that propositions are more likely to be recalled if they fit better in the story, that intrusion (i.e., false recall) is more likely for propositions that fit better in the story, and that these latter two effects increase over retention time. All of these effects have also been found empirically.

Second, the DSS model was extended with a process simulating the resolution of ambiguous pronouns. In the pronoun resolution model, a statement containing an ambiguous pronoun is turned into a microworld proposition when a discourse entity is chosen to serve as antecedent. The ambiguous statement can be represented by a vector in situation space, just like normal propositions. This vector is a combination of all vectors representing a possible outcome of the pronoun resolution process. This combination is modified by focusing, such that the vector for the ambiguous statement becomes closer to the vector resulting from choosing the most focused entity as the pronoun's referent.

A decision among possible referents is forced by making the vector ‘fall’ towards the situation space regions corresponding to disambiguated statements. This process is affected by the DSS model’s inference process, thereby implementing the effect of context information on pronoun resolution. The pronoun resolution model simulates how the initial interpretation of a pronoun depends on focus, but can be overridden by context information that is inconsistent with the focus. Moreover, it can account for empirical data regarding reading times and error rates, and explains how these are affected by focusing, context informativeness, and depth-of-processing.

As a third extension to the DSS model, an attempt was made to supply it with a textbase-level representation by training a recurrent neural network to transform sentences into the DSS vectors representing the situations described by the sentences. For this task, a microlanguage was developed to describe microworld situations. The network learned reasonably well to process these sentences. The intermediate representation that resulted was shown to combine surface text, propositional, and situational aspects of discourse in a single level of representation.

6.2 Conclusions

Our aim was to develop a simple, computationally sound model that predicts and explains empirical data. This seems to have been achieved. The DSS model's mathematical basis guarantees that it is computationally sound. The model is simple in the sense that it has only one free parameter and its equations follow directly from just a few, simplifying assumptions. Since design decisions were never made with any particular empirical data in mind, the fact that the model's results correspond to empirical data is an emergent property. Therefore, the model can be said to not only predict empirical data, but also to explain how the data arises. Moreover, we wanted the model to be able to handle different inputs, and indeed it can process any story taking place in the microworld. Also, the DSS model can generalize to several tasks, as is shown by extending it to simulate story retention and pronoun resolution.

6.2.1 Relation to theories of discourse comprehension

The model was designed to transform a given story into the trajectory that is most likely according to the model's world knowledge, with precision controlled by a depth-of-processing parameter. This is in accordance with the hypothesis that whether or not an inference is made does not directly depend on the type of inference, but on the availability of relevant knowledge, the extent to which the inference contributes to the story's coherence, and the reader's goals. Noordman, Vonk, and Kempff (1992), as well as Vonk and Noordman (1990), present empirical evidence supporting this view.

As an emergent property, the model's inference process turns out to generally result in increasing coherence, without coherence having any influence on the process. Inferences result in stronger coherence of the story representation at the situational level. For a large part, this is in line with the constructionist theory of inferencing (Graesser, Singer, & Trabasso, 1994) and not with the minimalist theory (McKoon & Ratcliff, 1992), which claims that elaborate situational representations are normally not made at all. However, the constructionist theory claims that inferences result from a search for coherence, while these roles are reversed in the DSS model, which views increase in coherence as a by-product of inferencing.

The Distributed Situation Space model and its extensions present a picture of discourse comprehension that is quite different from the view that has been prevalent since Kintsch and Van Dijk's (1978) influential paper. Like Golden and Rumelhart (1993), we have put central the situation model and its relation to the reader's knowledge instead of focusing on the propositional textbase and its relation to the text. The model does not deal with propositional structures but views propositions as facts about a microworld and explains how general knowledge shapes the interpretation of the incoming facts. In our view, understanding a text comes down to constructing a situational representation. Levels of representation prior to the situation model exist only because they are useful for transforming a text into a situation model. Although, at one of these levels, the units of meaning at which the text is represented may be proposition-like, there is no need for propositional predicate-argument structures.

6.2.2 Limitations

The model's focus on knowledge and higher-level mental representations entails two main drawbacks. First, a model dealing with world knowledge is necessarily constricted to a microworld if it is to avoid including knowledge subjectively and only on an ad hoc basis. The problem with any microworld is, of course, its small size. It is still an open question to what extent the model can be scaled up to handle more realistic amounts of knowledge. Since world knowledge is immediately available to the model in the form of the propositions' vector representations, it does not need to search through a knowledge base. Therefore, there seems to be no a priori reason to expect the model to break down when much more knowledge is included. However, implementing this knowledge may be problematic because of technical limitations. Training very large Self-Organizing Maps could turn out to be too laborious, while smaller maps result in an inaccurate implementation.

The model's second limitation is its lack of realistic input. The sequences of facts it processes are represented at a situational level, while the text from which these facts originate is ignored. In fact, the model's input might as well be based on something other than a text, for instance a movie. To the model, this makes no difference at all. Whatever the form of realistic input, however, it needs to be transformed into the situational vector representation. Section 5.3 showed that a recurrent neural network can be trained to transform word sequences into corresponding situation vectors. However, texts contain much additional information in the form of linguistic cues, which can influence the comprehension process but cannot be handled by the current model.

For example, in the DSS model, the temporal order of story situations is the same as the order in which they enter the model. As long as there are no linguistic cues, this constraint makes sense. Kamp and Rohrer (1983) and Hinrichs (1986) argue that the temporal and textual orders of story events are by default assumed to be the same. However, a connective or verb inflection might signal that the statement being processed describes an event that took place *before* a previously read situation. If the time step index of the new situation is somehow known, it could be inserted into the story trajectory at its appropriate place and be processed by the DSS model like any other situation. Finding the correct time step index, however, requires the use of a linguistic cue, probably in combination with the model's temporal world knowledge. If the model is to process realistic, textual input, it should be able to deal with such cues.

Pronouns are another source of textual information. It was shown how the DSS model can easily be extended to simulate pronoun resolution, but this is only a tiny first step towards turning the story comprehension model into a text comprehension model.

References

Albrecht, J.E., & Myers, J.L. (1991). Effects of centrality on retrieval of text-based concepts. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*, 932-939.

Albrecht, J.E., & Myers, J.L. (1995). Role of context in accessing distant information during reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*, 1459-1468.

Altmann, G.T.M. (1997). *The ascent of Babel: An exploration of language, mind, and understanding*. Oxford, UK: Oxford University Press.

Arnold, J.E., Eisenband, J.G., Brown-Schmidt, S., & Trueswell, J.C. (2000). The rapid use of gender information: evidence of the time course of pronoun resolution from eyetracking. *Cognition, 76*, B13-B26.

Ballard, D.H. (1997). *An introduction to natural computation*. Cambridge, MA: MIT Press.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B, 36*, 192-236.

Bestgen, Y., & Vonk, W. (1995). The role of temporal segmentation markers in discourse processing. *Discourse Processes, 19*, 385-406.

Bischofshausen, S. (1985). Developmental differences in schema dependency for temporally ordered story events. *Journal of Psycholinguistic Research, 14*, 543-556.

Bower, G.H., Black, J.B., & Turner, T.J. (1979). Scripts in memory for text. *Cognitive Psychology, 11*, 177-220.

Cacciari, C., & Glucksberg, S. (1994). Understanding figurative language. In M.A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 447-478). San Diego, CA: Academic Press.

Caramazza, A., Grober, E., Garvey, C., & Yates, J. (1977). Comprehension of anaphoric pronouns. *Journal of Verbal Learning and Verbal Behavior, 16*, 601-609.

Charniak, E. (1983). Passing markers: A theory of contextual influence in language comprehension. *Cognitive Science, 7*, 171-190.

Cressie, N.A.C. (1991). *Statistics for spatial data*. New York: John Wiley & Sons.

Dell, G.S., McKoon, G., & Ratcliff, R. (1983). The activation of antecedent information during the processing of anaphoric reference in reading. *Journal of Verbal Learning and Verbal Behavior, 22*, 121-132.

Desai, R. (2002). *Modeling interaction of syntax and semantics in language acquisition*. Unpublished doctoral dissertation, Indiana University, Bloomington, IN.

Dijkstra, T., & De Smedt, K. (1996). Computer models in psycholinguistics: an introduction. In T. Dijkstra & K. de Smedt (Eds.), *Computational Psycholinguistics* (pp. 3-23). London: Taylor & Francis.

Dirac, P.A.M. (1963). The evolution of the physicist's picture of nature. *Scientific American, 208*, 45-53.

Dormand, J.R., & Prince, P.J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics, 6*, 19-26.

Duffy, S.A., Shinjo, M., & Myers, J.L. (1990). The effect of encoding task on memory for sentence pairs varying in causal relatedness. *Journal of Memory and Language, 29*, 27-42.

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

Elman, J.L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition, 48*, 71-99.

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

Fletcher, C.R., & Bloom, C.P. (1988). Causal reasoning in the comprehension of simple narrative texts. *Journal of Memory and Language, 27*, 235-244.

Fletcher, C.R., & 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., & Vonk, W. (2003a). A model for knowledge-based pronoun resolution. In F. Detje, D. Dörner, & H. Schaub (Eds.), *The logic of cognitive systems: Proceedings of the fifth international conference on cognitive modeling* (pp. 245-246). Bamberg, Germany: Universitäts-Verlag.

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

Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003c). *Modeling multiple levels of text representation*. Manuscript submitted for publication.

Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2003d). *The effect of focus and knowledge on the resolution of ambiguous pronouns: a computational model*. Manuscript submitted for publication.

Frank, S.L., Koppen, M., Noordman, L.G.M., & Vonk, W. (2004). *[Discourse comprehension models: a critical analysis]*. Manuscript in preparation.

Gaddy, M.L., Van den Broek, P., & Sung, Y. (2001). The influence of text cues on the al-location of attention during reading. In T. Sanders, J. Schilperoord, & W. Spooren (Eds.), *Text representation: linguistic and psycholinguistic aspects* (pp. 89-110). Amsterdam: John Benjamins.

Garnham, A., Oakhill, J., & Cruttenden, H. (1992). The role of implicit causality and gender cue in the interpretation of pronouns. *Language and Cognitive Processes, 7*, 231-255.

Garnham, A., Traxler, M., Oakhill, J., & Gernsbacher, M.A. (1996). The locus of implicit causality effects in comprehension. *Journal of Memory and Language, 35*, 517-543.

Garrod, S.C., & Sanford, A.J. (1994). Resolving sentences in a discourse context: How discourse representation affects language understanding. In M.A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 675-698). San Diego, CA: Academic Press.

Garvey, C., Caramazza, A., & Yates, J. (1974). Factors influencing assignment of pronoun antecedents. *Cognition, 3*, 227-243.

Gernsbacher, M.A., & Hargreaves, D.J. (1988). Accessing sentence participants: The advantage of first mention. *Journal of Memory and Language, 27*, 699-717.

Gibbs, R.W., Jr. (1994). Figurative thought and figurative language. In M.A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 441-446). San Diego, CA: Academic Press.

Goetz, E.T., Anderson, R.C., & Schallert, D.L. (1981). The representation of sentences in memory. *Journal of Verbal Learning and Verbal Behavior, 20*, 369-385.

Golden, R.M. (1993). Stability and optimization analyses of the generalized brain-state-in-a-box neural network model. *Journal of Mathematical Psychology, 37*, 282-298.

Golden, R.M. (1996). *Mathematical methods for neural network analysis and design*. Cambridge, MA: MIT Press.

Golden, R.M., & Rumelhart, D.E. (1993). A parallel distributed processing model of story comprehension and recall. *Discourse Processes, 16*, 203-237.

Golden, R.M., Rumelhart, D.E., Strickland, J., & Ting, A. (1994). Markov random fields for text comprehension. In D.S. Levine & M. Aparicio (Eds.), *Neural networks for knowledge representation and inference* (pp. 283-309). Hillsdale, NJ: Erlbaum.

Golding, J.M., Millis, K.M., Hauselt, J., & Sego, S.A. (1995). The effect of connectives and causal relatedness on text comprehension. In R.F. Lorch & E.J. O'Brien (Eds.), *Sources of coherence in reading* (pp. 127-143). Hillsdale, NJ: Erlbaum.

Goldman, S.R., & Varnhagen, C.K. (1986). Memory for embedded and sequential story structures. *Journal of Memory and Language, 25*, 401-418.

Gordon, P.C., & Scearce, K.A. (1995). Pronominalization and discourse coherence, discourse structure and pronoun interpretation. *Memory & Cognition, 23*, 313-323.

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

Greene, S.B., McKoon, G., & Ratcliff, R. (1992). Pronoun resolution and discourse models. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*, 266-283.

Guha, A., & Rossi, J.-P. (2001). Convergence of the integration dynamics of the Construction-Integration model. *Journal of Mathematical Psychology, 45*, 355-369.

Hinrichs, E. (1986). Temporal anaphora in discourses of English. *Linguistics and Philosophy, 9*, 63-82.

Hinton, G.E., McClelland, J.L., & Rumelhart, D.E. (1986). Distributed representations. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing. Volume 1: Foundations* (pp. 77-109). Cambridge, MA: MIT Press.

Jacobs, A.M., & Grainger, J. (1994). Models of visual word recognition — Sampling the state if the art. *Journal of Experimental Psychology: Human Perception and Performance, 20*, 1311-1334.

Kamp, H. (1981). A theory of truth and semantic representation. In J.A.G. Groenendijk, T.M.V. Janssen, & M.B.J. Stokhof (Eds.), *Formal methods in the study of language, Part 1* (pp. 277-322). Amsterdam: Mathematisch Centrum.

Kamp, H., & Rohrer, C. (1983). Tense in texts. In R. Bäuerle, C. Schwarze, & A. von Stechow (Eds.), *Meaning, use and interpretation of language* (pp. 250-269). Berlin: de Gruyter.

Kaup, B., & Zwaan, R.A. (2003). Effects of negation and situational presence on the accessibility of text information. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*, 439-446.

Keenan, J.M., Potts, G.R., Golding, J.M., & Jennings, T.M. (1990). Which elaborative inferences are drawn during reading? A question of methodologies. In D.A. Balota, G.B. Flores d'Arcais, & K. Rayner (Eds.), *Comprehension processes in reading* (pp. 447-464). Hillsdale, NJ: Erlbaum.

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

Kintsch, W. (1992). How readers construct situation models for stories: The role of syntactic cues and causal inferences. In A.F. Healy, S.M. Kosslyn, & R.M. Shiffrin (Eds.), *From learning processes to cognitive processes: Essays in honor of William K. Estes, Volume 2* (pp. 261-278). Hillsdale, NJ: Erlbaum.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. Cambridge, UK: Cambridge University Press.

Kintsch, W. (2000). Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review, 7*, 257-266.

Kintsch, W. (2001). Predication. *Cognitive Science, 25*, 173-202.

Kintsch, W., & Bowles, A.R. (2002). Metaphor comprehension: What makes a metaphor difficult to understand?. *Metaphor and Symbol, 17*, 249-262.

Kintsch, W., & Van Dijk, T.A. (1978). Toward a model of text comprehension and production. *Psychological Review, 85*, 363-394.

Kintsch, W., & Welsch, D.M., Schmalhofer, F., & Zimny, S. (1990). Sentence memory: Theoretical analysis. *Journal of Memory and Language, 29*, 133-159.

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

Landauer, T.K., & Dumais, S.T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211-240.

Langston, M.C., & Trabasso, T. (1999). Modeling causal integration and availability of information during comprehension of narrative texts. In H. van Oostendorp & S.R. Goldman (Eds.), *The construction of mental representations during reading* (pp. 29-69). Mahwah, NJ: Erlbaum.

Langston, M.C., Trabasso, T., & Magliano, J.P. (1999). A connectionist model of narrative comprehension. In A. Ram & K. Moorman (Eds.), *Understanding language understanding: Computational models of reading* (pp. 181-226). Cambridge, MA: MIT Press.

Leonard, C.L., Waters, G.S., & Caplan, D. (1997a). The influence of contextual information on the resolution of ambiguous pronouns by younger and older adults. *Applied Psycholinguistics, 18*, 293-317.

Leonard, C.L., Waters, G.S., & Caplan, D. (1997b). The use of contextual information by right brain-damaged individuals in the resolution of ambiguous pronouns. *Brain and Language, 57*, 309-342.

Long, D.L., & De Ley, L. (2000). Implicit causality and discourse focus: The interaction of text and reader characteristics in pronoun resolution. *Journal of Memory and Language, 42*, 545-570.

Luftig, R.L. (1982). Effects of paraphrase and schema on intrusions, normalizations, and recall of thematic prose. *Journal of Psycholinguistic Research, 11*, 369-380.

Lutz, M.F., & Radvansky, G.A. (1997). The fate of completed goal information in narrative comprehension. *Journal of Memory and Language, 36*, 293-310.

Magliano, J.P., Zwaan, R.A., & Graesser, A. (1999). The role of situational continuity in narrative understanding. In H. van Oostendorp & S.R. Goldman (Eds.), *The construction of mental representations during reading* (pp. 219-245). Mahwah: Erlbaum.

McDonald, J.T., & MacWhinney, B. (1995). The time course of anaphor resolution: Effects of implicit verb causality and gender. *Journal of Memory and Language, 34*, 543-566.

McKoon, G., & Ratcliff, R. (1980). The comprehension processes and memory structures involved in anaphoric reference. *Journal of Verbal Learning and Verbal Behavior, 19*, 668-682.

McKoon, G., & Ratcliff, R. (1992). Inference during reading. *Psychological Review, 99*, 440-466.

Miikkulainen, R. (1993). *Subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory*. Cambridge, MA: MIT Press.

Miikkulainen, R., & Dyer, M.G. (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science, 15*, 343-399.

Millis, K.K., Graesser, A.C., & Haberlandt, K. (1993). The impact of connectives on the memory for expository texts. *Applied Cognitive Psychology, 7*, 317-339.

Millis, K.K., & Just M.A. (1994). The influence of connectives on sentence comprehension. *Journal of Memory and Language, 33*, 128-147.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Murray, J.D. (1995). Logical connectives and local coherence. In F. Lorch & E.J. O'Brien (Eds.), *Sources of coherence in reading* (pp. 107-125). Hillsdale, NJ: Erlbaum.

Murray, J.D. (1997). Connectives and narrative text: The role of continuity. *Memory & Cognition, 25*, 227-236.

Myers, J.L., & O'Brien, E.J. (1998). Accessing the discourse representation during reading. *Discourse Processes, 26*, 131-157.

Myers, J.L., Shinjo, M., & Duffy, S.A. (1987). Degree of causal relatedness and memory. *Journal of Memory and Language, 26*, 453-465.

Noordman, L.G.M., & Vonk, W. (1992). Reader's knowledge and the control of inferences in reading. *Language and Cognitive Processes, 7*, 373-391.

Noordman, L.G.M., & Vonk, W. (1998). Discourse comprehension. In A.D. Friederici (Ed.), *Language comprehension: a biological perspective* (pp. 229-262). Berlin: Springer.

Noordman, L.G.M., Vonk, W., & Kempff, H.J. (1992). Causal inferences during the reading of expository texts. *Journal of Memory and Language, 31*, 573-590.

Norvig, P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science, 13*, 569-620.

Oakhill, J., Garnham, A., & Vonk, W. (1989). The on-line construction of discourse models. *Language and Cognitive Processes, 4*, SI263-SI286.

O'Brien, E.J. (1987). Antecedent search processes and the structure of text. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*, 278-290.

O'Brien, E.J., Albrecht, J.E., Hakala, C.M., & Rizzella, M.L. (1995). Activation and suppression of antecedents during reinstatement. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*, 626-634.

Perfetti, C.A., & Britt, M.A. (1995). Where do propositions come from?. In C.A. Weaver, S. Mannes, & C.R. Fletcher (Eds.), *Discourse comprehension: Essays in honor of Walter Kintsch* (pp. 11-34). Hillsdale, NJ: Erlbaum.

Pollack, J.B. (1995). The induction of dynamical recognizers. In R.F. Port & T. van Gelder (Eds.), *Mind as motion: explorations in the dynamics of cognition* (pp. 1-43). Cambridge, MA: MIT Press.

Ratcliff, R., & McKoon, G. (1978). Priming in item recognition: evidence for the propositional structure of sentences. *Journal of Verbal Learning and Verbal Behavior, 17*, 403-417.

Rodenhausen, H. (1992). Mathematical aspects of Kintsch's model of discourse comprehension. *Psychological Review, 99*, 547-549.

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

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing. Volume 1: Foundations* (pp. 318-362). Cambridge, MA: MIT Press.

Sanders, T.J.M., & Noordman, L.G.M. (2000). The role of coherence relations and their linguistic markers in text processing. *Discourse Processes, 29*, 37-60.

Schmalhofer, F., McDaniel, M.A., & Keefe, D. (2002). A unified model for predictive and bridging inferences. *Discourse Processes, 33*, 105-132.

Schmidt, A.M.G. (1963). *Jip en Janneke: eerste boek [Bob and Jilly: first book]*. Amsterdam: De Arbeiderspers.

Singer, M. (1994). Discourse inference processes. In M.A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 479-515). San Diego, CA: Academic Press.

Singer, M. (1996). Comprehending consistent and inconsistent causal text sequences: A Construction-Integration analysis. *Discourse Processes, 21*, 1-21.

Smith, D.A., & Graesser, A.C. (1981). Memory for actions in scripted activities as a function of typicality, retention interval, and retrieval task. *Memory & Cognition, 9*, 550-559.

Stewart, A.J., & Gosselin, F. (2000). A simple categorisation model of anaphor resolution. In L.R. Gleitman & A.K. Joshi (Eds.), *Proceedings of the twenty-second annual conference of the cognitive science society* (pp. 930-935). Philadelphia: University of Pennsylvania.

Stewart, A.J., Pickering, M.J., & Sanford, A.J. (2000). The time course of the influence of implicit causality information: Focusing versus integration accounts. *Journal of Memory and Language, 42*, 423-443.

St.John, M.F. (1992). The Story Gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science, 16*, 271-306.

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

St.John, M.F., & McClelland, J.L. (1992). Parallel constraint satisfaction as a comprehension mechanism. In R.G. Reilly & N.E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 97-136). Hove, UK: Erlbaum.

Suh, S., & Trabasso, T. (1993). Inferences during on-line processing: Converging evidence from discourse analysis, talk-aloud protocols, and recognition priming. *Journal of Memory and Language, 32*, 279-301.

Tabor, W., Juliano, C., & Tanenhaus, M.K. (1997). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes, 12*, 211-271.

Tabor, W., & Tanenhaus, M.K. (1999). Dynamical models of sentence processing. *Cognitive Science, 23*, 491-515.

Tapiero, I., & Denhière, G. (1995). Simulating recall and recognition by using Kintsch's Construction-Integration model. In C.A. Weaver, S. Mannes, & C.R. Fletcher (Eds.), *Discourse comprehension: Essays in honor of Walter Kintsch* (pp. 211-232). Hillsdale, NJ: Erlbaum.

Till, R.E., Mross, E.F., & Kintsch, W. (1988). Time course of priming for associate and inference words in a discourse context. *Memory & Cognition, 16*, 283-298.

Trabasso, T., Secco, T., & Van den Broek, P. (1984). Causal cohesion and story coherence. In H. Mandl, N.L. Stein, & T. Trabasso (Eds.), *Learning and comprehension of text* (pp. 83-111). Hillsdale, NJ: Erlbaum.

Trabasso, T., & Sperry, L.L. (1985). Causal relatedness and importance of story events. *Journal of Memory and Language, 24*, 595-611.

Trabasso, T., & Van den Broek, P. (1985). Causal thinking and the representation of narrative events. *Journal of Memory and Language, 24*, 612-630.

Turner, A., & Greene, E. (1978). The construction and use of a propositional text base. *Catalog of Selected Documents in Psychology, 8, 58*, (MS No. 1713).

Van den Broek, P. (1994). Comprehension and memory of narrative texts: Inferences and coherence. In M.A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 539-588). San Diego, CA: Academic Press.

Van den Broek, P., Risden, K., Fletcher, C.R., & Thurlow, R. (1996). A "landscape" view of reading: Fluctuating patterns of activation and the construction of a stable memory representation. In B.K. Britton & A.C. Graesser (Eds.), *Models of Understanding Text* (pp. 165-187). Mahwah, NJ: Erlbaum.

Van den Broek, P., Young, M., Tzeng, Y., & Linderholm, T. (1999). The landscape model of reading: Inferences and the online construction of a memory representation. In H. van Oostendorp & S.R. Goldman (Eds.), *The construction of mental representations during reading* (pp. 71-98). Mahwah, NJ: Erlbaum.

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

Van Gelder, T., & Port, R.F. (1995). It's about time: An overview of the dynamical approach to cognition. In R.F. Port & T. van Gelder (Eds.), *Mind as motion: explorations in the dynamics of cognition* (pp. 1-43). Cambridge, MA: MIT Press.

Varnhagen, C.K., Morrison, F.J., & Everall, R. (1994). Age and schooling effects in story recall and story production. *Developmental Psychology, 30*, 969-979.

Vonk, W. (1985). The immediacy of inferences in the understanding of pronouns. In G. Rickheit & H. Strohner (Eds.), *Inferences in text processing* (pp. 205-218). Amsterdam: Elsevier Science.

Vonk, W., & Noordman, L.G.M. (1990). On the control of inferences in text understanding. In D.A. Balota, G.B. Flores d'Arcais, & K. Rayner (Eds.), *Comprehension processes in reading* (pp. 447-464). Hillsdale, NJ: Erlbaum.

Weeber, M. (1996). *On Modeling Text Processing*. Unpublished master's thesis, University of Nijmegen, Nijmegen, The Netherlands.

Zwaan, R.A. (1999). Embodied cognition, perceptual symbols, and situation models. *Discourse Processes, 28*, 81-88.

Zwaan, R.A., & Radvansky, G.A. (1998). Situation models in language comprehension and memory. *Psychological Bulletin, 123*, 162-185.

Zwaan, R.A., Stanfield, R.A., & Yaxley, R.H. (2002). Language comprehenders mentally represent the shapes of objects. *Psychological Science, 13*, 168-171.

Appendices

Appendix A provides details of some of the simulations that were run. The input to the Resonance model of Section 2.1 is given in Appendix A.1. In Section 5.3, different groups of sentence pairs were compared to investigate the nature of the sentences' vector representations. These sentence pairs are shown in Appendix A.2.

Mathematical details are provided in Appendix B. First, convergence of the Resonance model is proven in Section B.1. Second, Section B.2 discusses how local probabilities are computed for the Golden and Rumelhart and DSS models. Third, Section B.3 proves that the DSS model always converges.

Appendix

A Simulation details

A.1 The Resonance model

The captain text used by Albrecht and Myers (1995) and stated in Myers and O'Brien (1998, Table 1) was parsed into 23 concepts (Table A1) and 63 propositions (Table A2). Together with 17 sentence markers, these formed the 103 nodes of the text network. Eleven of these are critical items: They are related to the concept INVENTORY and are expected to be reinstated into working memory by the 15th sentence of the text.

Table A1: All concepts extracted from Albrecht and Myers' captain text. The critical concept is indicated by ⋆.

	label	concept
	C1	CRUISE
	C2	SHIP
	C3	CAPTAIN
	C4	OFFICE
	C5	PAPERWORK
⋆	C6	INVENTORY
	C7	LEAVE
	C8	CHAIR
	C9	DESK
	C10	PASSENGERS
	C11	THEFT
	C12	FORMS
	C13	INVESTIGATION
	C14	THIEF
	C15	COMPLAINTS
	C16	MINUTES
	C17	STAFF_MEMBER
	C18	MASTER_KEY
	C19	CABINS
	C20	NUMBER_OF_SUSPECTS
	C21	CREW_MEMBERS
	C22	PURSER
	C23	SHORE

Table A2: All propositions and sentence markers, in order of sentence of occurrence, extracted from Albrecht and Myers' captain text. The critical propositions are indicated by ⋆. The markers of sentences that contain a critical proposition are themselves critical items. In the alternative text, which did not mention the reinstating concept DESK, proposition P56 reads SAT(CAPTAIN) in stead of SAT(CAPTAIN,at:LARGE(DESK)).

	sent.		label	proposition
	S1		P1	ENDING(CRUISE)
			P2	DOCK(SHIP)
			P3	SOON(P2)
			P4	AND(P1,P3)
	S2		P5	SAT(CAPTAIN,in:OFFICE)
			P6	FINISH(CAPTAIN,PAPERWORK)
			P7	TRIES(CAPTAIN,P6)
			P8	FRANTICALLY(P7)
⋆	S3	⋆	P9	OF(INVENTORY,SHIP)
			P10	MUST_DO(CAPTAIN,P9)
			P11	BEGIN(CAPTAIN,LEAVE)
			P12	BEFORE(P10,P11)
⋆	S4		P13	FINED(CAPTAIN)
			P14	HEAVILY(P13)
		⋆	P15	COMPLETE(CAPTAIN,INVENTORY)
			P16	EARLIER(CRUISE)
			P17	DID_NOT(CAPTAIN,P15,on:P16)
			P18	BECAUSE(P14,P17)
	S5		P19	LARGE(DESK)
			P20	PULLED_UP(CAPTAIN,CHAIR)
			P21	SAT(CAPTAIN,at:P19)
			P22	AND(P20,P21)
⋆	S6	⋆	P23	START(CAPTAIN,INVENTORY)
			P24	ARRIVED(PASSENGERS)
			P25	REPORT(PASSENGERS,THEFT)
			P26	IN_ORDER_TO(P24,P25)
			P27	BEFORE(P23,P26)
⋆	S7	⋆	P28	COMPLETE(CAPTAIN,INVENTORY)
			P29	LATER(P28)
			P30	MUST(CAPTAIN,P29)

continued on next page

continued from previous page

	sent.		label	proposition
⋆	S8	⋆	P31	KIND_OF(FORMS,INVENTORY)
			P32	COVERED(P31,DESK)
			P33	LEFT(CAPTAIN,P32)
			P34	BEGAN(CAPTAIN,INVESTIGATION)
			P35	CATCH(CAPTAIN,THIEF)
			P36	IN_ORDER_TO(P34,P35)
			P37	AND(P33,P36)
	S9		P38	REVIEWED(CAPTAIN,COMPLAINTS)
			P39	CAREFULLY(P38)
	S10		P40	WAS(STAFF_MEMBER, THIEF)
			P41	SURE(CAPTAIN,P40)
			P42	AFTER(MINUTES,P41)
	S11		P43	OF(CABINS,PASSENGERS)
			P44	OPENS(MASTER_KEY,P43)
			P45	ACCESS(THIEF,to:P44)
	S12		P46	REDUCED(P45,NUMBER_OF_SUSPECTS)
			P47	GREATLY(P46)
	S13		P48	QUESTIONED(CAPTAIN,CREW_MEMBERS)
			P49	OF(PURSER,SHIP)
			P50	WAS(P49,THIEF)
			P51	SURE(CAPTAIN,P50)
			P52	AFTER(P51,P48)
	S14		P53	LOCKED_UP(PURSER)
			P54	WITHIN(P53,MINUTES)
	S15		P55	RETURNED(CAPTAIN,to:OFFICE)
			P56	SAT(CAPTAIN[,at:P19])
			P57	AND(P55,P56)
	S16		P58	HAPPY(CAPTAIN)
			P59	DONE(CAPTAIN,with:CRUISE)
			P60	BECAUSE(P58,P59)
	S17		P61	KIND_OF(LEAVE,SHORE)
			P62	START(CAPTAIN,P61)
			P63	READY(CAPTAIN,to:P62)

A.2 A textbase-level representation for the DSS model

In order to investigate the nature of the intermediate representations of sentences developed in Section 5.3, distances between vectors are compared. Table A3 shows the three sets of sentence pairs used for this comparison.

Table A3: Three sets of sentence pairs. Two sentences of a pair from the 'surface different' set differ only in surface text and not propositionally. Sentences of a pair from the 'textbase different' set differ propositionally but describe identical situations. Sentences of a pair from the 'situation different' set differ both propositionally and situationally.

surface different	
Bob and Jilly play soccer	Jilly and Bob play soccer
Bob and Jilly play soccer outside	Jilly and Bob play soccer outside
Bob and Jilly play hide-and-seek	Jilly and Bob play hide-and-seek
Bob and Jilly play hide-and-seek inside	Jilly and Bob play hide-and-seek inside
Bob and Jilly play a_computer_game	Jilly and Bob play a_computer_game
Bob and Jilly play a_computer_game inside	Jilly and Bob play a_computer_game inside
Bob and Jilly play with_the_dog	Jilly and Bob play with_the_dog
Bob and Jilly play with_the_dog outside	Jilly and Bob play with_the_dog outside
textbase different	
Bob play soccer	Jilly play soccer
Bob play hide-and-seek	Jilly play hide-and-seek
Bob play soccer	Bob play soccer outside
Jilly play soccer	Jilly play soccer outside
Bob play a_computer_game	Bob play a_computer_game inside
Jilly play a_computer_game	Jilly play a_computer_game inside
Bob and Jilly play soccer	Bob and Jilly play soccer outside
Jilly and Bob play soccer	Jilly and Bob play soccer outside
Bob and Jilly play a_computer_game	Bob and Jilly play a_computer_game inside
Jilly and Bob play a_computer_game	Jilly and Bob play a_computer_game inside
situation different	
Bob play a_computer_game	Jilly play a_computer_game
Bob play with_the_dog	Jilly play with_the_dog
Bob play hide-and-seek	Bob play hide-and-seek inside
Jilly play hide-and-seek	Jilly play hide-and-seek inside
Bob play with_the_dog	Bob play with_the_dog outside
Jilly play with_the_dog	Jilly play with_the_dog outside
Bob and Jilly play hide-and-seek	Bob and Jilly play hide-and-seek inside
Jilly and Bob play hide-and-seek	Jilly and Bob play hide-and-seek inside
Bob and Jilly play with_the_dog	Bob and Jilly play with_the_dog outside
Jilly and Bob play with_the_dog	Jilly and Bob play with_the_dog outside

B Mathematical details

B.1 Convergence of the Resonance model

The vector $N = (n_p, n_q, \ldots)$ is defined as the sum of the columns of connectivity matrix W, which equals the sum of its rows because the matrix is symmetric. Now define the matrix $\mathcal{N}$ such that its diagonal is formed by $(n_p^{-1}, n_q^{-1}, \ldots)$ and all other elements are 0.

If the resonance process can be shown to converge with threshold parameter $\theta = 0$, it must also converge for larger values. Assuming that $\theta = 0$, the resonance algorithm is given by the pair of equations

$$\begin{aligned} X(c+1) &= X(c) + S(c)W \\ S(c) &= (1-\gamma)^{c-1}X(c)\mathcal{N}. \end{aligned}$$

Taken together, the expression for the resonance vector becomes

$$\begin{aligned} X(c+1) &= X(c) + (1-\gamma)^{c-1}X(c)\mathcal{N}W \\ &= X(c)\left(I + (1-\gamma)^{c-1}K\right) \end{aligned} \tag{B1}$$

where I is the identity matrix and matrix K is defined as $K = \mathcal{N}W$. Equation B1 shows that in every processing cycle, the resonance vector is multiplied by $I + (1-\gamma)^{c-1}K$. Since

$$\lim_{c\to\infty}(I + (1-\gamma)^{c-1}K) = I$$

for $0 < \gamma \leq 1$, the resonance values eventually no longer change for values of the decay rate parameter γ larger than 0.

B.2 Local probability

In the Golden and Rumelhart model, the expected value of a proposition p at time step t is computed using the *local* probability distribution $P_{p,t}$. This is the probability distribution of p_t given the values of all other propositions-at-time-steps. Equation 3.1 gives the ratio of two *global* probabilities, but the ratio of the corresponding local probabilities is easily shown to be the same.

Let $\bar{X}^*_{p,t}$ denote the collection of all values of the trajectory *except* that of p_t. The ratio of the local probabilities $P_{p,t}(x_1)$ and $P_{p,t}(x_2)$ equals

$$\frac{P_{p,t}(x_1)}{P_{p,t}(x_2)} = \frac{P(x_1|\bar{X}^*_{p,t})}{P(x_2|\bar{X}^*_{p,t})} = \frac{P(x_1, \bar{X}^*_{p,t})/P(\bar{X}^*_{p,t})}{P(x_2, \bar{X}^*_{p,t})/P(\bar{X}^*_{p,t})}$$
$$= \frac{P(x_1, \bar{X}^*_{p,t})}{P(x_2, \bar{X}^*_{p,t})} = \mathrm{e}^{Q(x_1, \bar{X}^*_{p,t}) - Q(x_2, \bar{X}^*_{p,t})}.$$

From Equation 3.3 it follows that

$$Q(x_1, \bar{X}^*_{p,t}) - Q(x_2, \bar{X}^*_{p,t}) = (x_1 - x_2)(b_p + X_{t-1}W_{\cdot p} + W_{p\cdot}X'_{t+1}).$$

Here, $W_{\cdot p}$ and $W_{p\cdot}$ are the pth column, respectively the pth row, of W. For this equation to be valid at every time step, the vectors X_0 and X_{T+1} are defined to consist of 0s only. We shall use the shorthand notation

$$\Delta Q_{p,t} = b_p + X_{t-1}W_{\cdot p} + W_{p\cdot}X'_{t+1}.$$

Note that $\Delta Q_{p,t}$ is a function of $\bar{X}^*_{p,t}$, but does not depend on the value x_1 or x_2 of p_t. The ratio of local probabilities can now more simply be written as

$$\frac{P_{p,t}(x_1)}{P_{p,t}(x_2)} = \mathrm{e}^{(x_1 - x_2)\Delta Q_{p,t}}. \tag{B2}$$

The local probability distribution can be derived from Equation B2. There are theoretically only two possible values, 0 and 1, for a proposition-at-a-time-step. Taking $x_1 = 0$ and $x_2 = 1$ in Equation B2, and using $P_{p,t}(0) + P_{p,t}(1) = 1$, leads to

$$P_{p,t}(1) = \frac{1}{1 + \mathrm{e}^{-\Delta Q_{p,t}}},$$

that is, the logistic function of $\Delta Q_{p,t}$. With 0 and 1 as the possible values, this probability that $x_{p,t} = 1$ equals the expected value of the local probability distribution of p_t.

For the DSS model, the local probability of SOM-cells-at-time-steps can be computed in a manner similar to the computation of the local probability of propositions-at-time-steps in the Golden and Rumelhart model. There are only three differences. First, proposition indices p are replaced by SOM-cell indices i. Second, all bias values b_i equal 0. Third, SOM-cells can theoretically have any value between 0 and 1. Consequently, the local probability $P_{i,t}$ is to be replaced by a probability density. Applying Equation B2 to this density, with $x_2 = 0$ and $x_1 = x$, $x \in [0,1]$, leads to the following equation for the density $P_{i,t}$:

$$P_{i,t}(x) = P_{i,t}(0)\mathrm{e}^{x\Delta Q_{i,t}}.$$

Being a probability density, $P_{i,t}$ has to integrate to unity over the interval $[0,1]$. Thus, for $\Delta Q_{i,t} \neq 0$,

$$\begin{aligned}\int_0^1 P_{i,t}(x)\mathrm{d}x &= P_{i,t}(0)\int_0^1 \mathrm{e}^{x\Delta Q_{i,t}}\mathrm{d}x \\ &= P_{i,t}(0)\left[(\Delta Q_{i,t})^{-1}\mathrm{e}^{x\Delta Q_{i,t}}\right]_0^1 \\ &= P_{i,t}(0)(\Delta Q_{i,t})^{-1}(\mathrm{e}^{\Delta Q_{i,t}} - 1) = 1,\end{aligned}$$

showing that $P_{i,t}(0) = \Delta Q_{i,t}\left(\mathrm{e}^{\Delta Q_{i,t}} - 1\right)^{-1}$ and so

$$P_{i,t}(x) = \frac{\Delta Q_{i,t}\mathrm{e}^{x\Delta Q_{i,t}}}{\mathrm{e}^{\Delta Q_{i,t}} - 1}.$$

If $\Delta Q_{i,t}$ approaches zero, this density approaches the uniform density $P_{i,t}(x) = 1$ on the interval $[0,1]$. This density also results directly from applying the above argument to the case $\Delta Q_{i,t} = 0$.

Inspection of the expression for $P_{i,t}$ makes clear that the maximum probability density is always obtained for one of the extreme values: for $x_{i,t} = 0$ if $\Delta Q_{i,t} < 0$, and for $x_{i,t} = 1$ if $\Delta Q_{i,t} > 0$. This is why the inference model described in Section 4.4 lets each $x_{i,t}$ approach either 0 or its maximum value.

For $\Delta Q_{i,t} \neq 0$, the expected value of $x_{i,t}$ is obtained through integration by parts:

$$\begin{aligned}E_{i,t}(\Delta Q_{i,t}) &= \int_0^1 xP_{i,t}(x)\mathrm{d}x \\ &= \frac{\Delta Q_{i,t}}{\mathrm{e}^{\Delta Q_{i,t}} - 1}\int_0^1 x\mathrm{e}^{x\Delta Q_{i,t}}\mathrm{d}x \\ &= \frac{\Delta Q_{i,t}}{\mathrm{e}^{\Delta Q_{i,t}} - 1}\left(\left[(\Delta Q_{i,t})^{-1}x\mathrm{e}^{x\Delta Q_{i,t}}\right]_0^1 - \int_0^1 (\Delta Q_{i,t})^{-1}\mathrm{e}^{x\Delta Q_{i,t}}\mathrm{d}x\right)\end{aligned}$$

$$
\begin{aligned}
&= \frac{\Delta Q_{i,t}}{\mathrm{e}^{\Delta Q_{i,t}} - 1} \left((\Delta Q_{i,t})^{-1} \mathrm{e}^{\Delta Q_{i,t}} - \left[(\Delta Q_{i,t})^{-2} \mathrm{e}^{x \Delta Q_{i,t}} \right]_0^1 \right) \\
&= \frac{1}{1 - \mathrm{e}^{-\Delta Q_{i,t}}} - \frac{1}{\Delta Q_{i,t}}.
\end{aligned}
$$

According to this expression $E_{i,t}(0) = \frac{1}{2}$ in the limit for $\Delta Q_{i,t}$ going to zero, corresponding to the expected value of the uniform density valid for $\Delta Q_{i,t} = 0$.

B.3 Convergence of the DSS model

Convergence of the DSS algorithm is proven by showing that there exists a so-called Lyapunov function for the system. A function $L(\bar{X})$ that associates a real value to any story trajectory $\bar{X}$ is a Lyapunov function if it has a minimum value, it is continuous and has continuous partial derivatives, and it decreases whenever the trajectory changes according to the model's Equation 4.9 (Ballard, 1997, chap. 5.3). Since any change in $\bar{X}$ decreases $L(\bar{X})$, which has a minimum value, the existence of such a function proves that the model converges.

It is not hard to show that a Lyapunov function for the model is

$$L(\bar{X}) = -\sum_{t=2}^{T} \sum_{i,j} x_{i,t-1} w_{ij} x_{j,t}.$$

First of all, L must have a minimum since it is a finite sum over finite values. Second, it clearly is continuous. Third, the derivative of L with respect to $x_{i,t}$ equals

$$
\begin{aligned}
\frac{\mathrm{d}L}{\mathrm{d}x_{i,t}} &= -\sum_{j} \left(w_{ji} x_{j,t-1} + w_{ij} x_{j,t+1} \right) \\
&= -\left(X_{t-1} W_{\cdot i} + W_{i\cdot} X'_{t+1} \right), \qquad \text{(B3)}
\end{aligned}
$$

which is also continuous. Fourth, $L(\bar{X})$ can be shown to decreases with any change in $\bar{X}$. Note that its derivative with respect to $x_{i,t}$ (Equation B3) equals $-\Delta Q_{i,t}$ from Equation 4.5. Therefore,

$$\mathrm{d}L = -\Delta Q_{i,t} \mathrm{d}x_{i,t}. \qquad \text{(B4)}$$

From Equations 4.4 and 4.9 it follows that $x_{i,t}$ increases whenever $\Delta Q_{i,t}$ is positive and decreases when $\Delta Q_{i,t}$ is negative. In other words, $\mathrm{d}x_{i,t}$ and $\Delta Q_{i,t}$

always have the same sign. By Equation B4, this means that dL is always negative, that is, $L(\bar{X})$ decreases with any change in $\bar{X}$, proving that it is a Lyapunov function and that the model converges.

Samenvatting

Zinnen staan zelden op zichzelf. Meestal maakt een zin deel uit van een tekst, en is daarom niet volledig te begrijpen zonder een verband te leggen met de andere zinnen. Onder normale omstandigheden doen lezers en luisteraars dit automatisch en zonder zich ervan bewust te zijn. Theorieën over dit mentale proces van tekstbegrip zijn het onderwerp van dit proefschrift. In het bijzonder wordt gekeken naar theorieën die zo volledig en gedetailleerd zijn dat ze kunnen worden uitgeschreven in de vorm van wiskundige formules of computerprogramma's. Dit soort theorieën wordt *computationele modellen* genoemd.

Acht modellen

Hoofdstuk 2 bespreekt zeven computationele modellen van tekstbegrip: het Resonance model, het Landscape model, Langston en Trabasso's model, het Construction-Integration model, het Predication model, het Sentence Gestalt model, en het Story Gestalt model. In Hoofdstuk 3 wordt het model van Golden en Rumelhart hier aan toegevoegd.

Het Resonance model (Myers & O'Brien, 1998) simuleert hoe het verwerken van een zin leidt tot activatie van concepten en proposities in de mentale representatie van de voorafgaande tekst, zoals die door de lezer is gevormd. Het model laat zien hoe 'oude' tekstelementen uit het werkgeheugen verdwijnen wanneer ze niets te maken hebben met de zin die gelezen wordt, en hoe ze weer in het werkgeheugen kunnen terugkeren door het lezen van een woord waarmee ze eerder samengingen. Hoewel het model dit proces met succes nabootst, lijdt het aan enkele tekortkomingen. Ten eerste zijn de parameterinstellingen waarschijnlijk niet tekstonafhankelijk. Ten tweede kunnen waarden van variabelen tot onrealistische hoogte stijgen tijdens het verwerkingsproces. Ten derde wordt de kennis van de lezer buiten beschouwing gelaten, ook al is die soms noodzakelijk voor het vinden van verbanden binnen een tekst.

De activatie van tekstelementen in het werkgeheugen leidt ertoe dat de tekst een spoor achterlaat in het episodisch geheugen. Het simpelste van de zeven modellen van Hoofdstuk 2, het Landscape model (Van den Broek, Risden, Fletcher, & Thurlow, 1996), berekent uit hypothetische activaties van concepten in het werkgeheugen, hoe sterk tekstelementen en relaties tussen de elementen worden vastgelegd in dit geheugenspoor. In feite is het model een implementatie van het algemeen aanvaarde idee dat tekstelementen beter worden onthouden naarmate ze vaker in het werkgeheugen voorkomen, en dat twee tekstelementen sterker met elkaar worden geassocieerd naarmate ze vaker samen voorkomen in het werkgeheugen. De sterkte van het Landscape model ligt in zijn eenvoud, wat tegelijk zijn belangrijkste zwakte is. Omdat de resultaten zo direct volgen uit de invoer, levert het model geen verrassende uitkomsten op en verklaart het weinig over het activeren van concepten en het vormen van een geheugenspoor.

Het model van Langston en Trabasso (1999) is het eerste van de besproken modellen waarin een rol is weggelegd voor de achtergrondkennis van de lezer, de zogeheten wereldkennis. Het model richt zich op oorzakelijke relaties tussen de gebeurtenissen in een verhaal. De invoer bestaat uit een lijst met mogelijke oorzakelijke verbanden tussen verhaalgebeurtenissen. Het model zou moeten berekenen welke van deze verbanden echt door een lezer worden gelegd. Het blijkt echter dat dit model niets anders voorspelt dan het ongewenste effect dat zinnen die eerder in de tekst staan sterker verbonden worden dan zinnen die later in de tekst voorkomen. Daar komt nog bij dat het model alle mogelijke oorzakelijke verbanden binnen een tekst als invoer moet ontvangen, wat verre van realistisch is.

Elk model waarin wereldkennis een rol speelt heeft problemen met de reusachtige hoeveelheid kennis die lezers in de strijd kunnen gooien om een tekst te begrijpen. Geen realistisch deel van al deze kennis kan worden geïmplementeerd . Verreweg het bekendste en invloedrijkste model van tekstbegrip, het Construction-Integration model (Kintsch, 1988, 1998), pakt dit probleem aan door ervan uit te gaan dat het verwerken van een zin in twee fasen verloopt. In de eerste fase, de constructiefase, worden elementen uit de zin geassocieerd met een paar elementen uit de wereldkennis van de lezer. Hoe sterker de relatie tussen tekst- en kenniselement, hoe groter de kans dat het kenniselement wordt uitgekozen. Alleen deze geassocieerde elementen hoeven te worden geïmplementeerd in het model. Andere zinnen uit de tekst (dat wil zeggen, de context) hebben geen enkele invloed op de constructiefase. Het is tijdens de tweede fase, integratie genaamd, dat een contextgevoelig proces irrelevante of inconsistente elementen verwijdert uit de verzameling van tekst- en kenniselementen. Wat overblijft, is de interpretatie van de tekst door de lezer.

Hoewel het Construction-Integration model geprezen kan worden voor het erkennen van het belang van kennis voor het begrijpen van teksten, doet het zijn reputatie geen eer aan. Afgezien van aanzienlijke technische tekortkomingen van het integratieproces, lijkt het basisidee ontoereikend. Een contextonafhankelijk constructieproces kan niet in het algemeen op de proppen komen met noodzakelijke associaties uit het kennisbestand van de lezer, tenzij het zoveel associaties produceert dat de constructiefase geen verklarende waarde meer heeft.

Het volgende model dat wordt besproken is het Predication model (Kintsch, 2001). Dit model is een uitbreiding op Latent Semantic Analysis (Landauer & Dumais, 1997), een methode voor het ontwikkelen van vectorrepresentaties van woorden. Overeenkomsten tussen vectoren gaan samen met betekenisovereenkomsten tussen de woorden die worden gerepresenteerd. Op die manier coderen de vectoren een deel van de woordbetekenissen. Het Predication model probeert te verklaren hoe deze woordvectoren kunnen worden gecombineerd tot representaties van proposities. Proposities hebben echter te maken met waarheidswaarden en die kunnen nooit volgen uit het simpelweg combineren van individuele woordbetekenissen. Het Predication model kan daarom wel woordvectoren aanpassen aan de context, maar niet tot representaties van proposities komen.

Het Sentence Gestalt model (St. John & McClelland, 1990, 1992) ontwikkelt wèl vectorrepresentaties voor proposities. In feite is het model een neuraal netwerk met terugkoppeling, dat wordt getraind om zinnen (in de vorm van series woorden) te verwerken en vragen te beantwoorden over de gebeurtenissen die door de zinnen worden beschreven. Tijdens het trainen van het netwerk ontwikkelt het vectorrepresentaties van de zinnen. Het Story Gestalt model (St. John, 1992), dat dezelfde architectuur heeft als het Sentence Gestalt model, leert verhalen (series proposities) verwerken en vragen beantwoorden over de gebeurtenissen in een verhaal. Om deze taak uit te voeren moet het een vectorrepresentatie van verhalen ontwikkelen.

Eenmaal getraind kan het Sentence Gestalt model een zin verwerken en ontbrekende, maar afleidbare, informatie invullen. Op dezelfde manier kan het Story Gestalt model zijn kennis van verhalen gebruiken om af te leiden welke gebeurtenissen moeten hebben plaatsgevonden, maar niet genoemd zijn in een verhaal. Het model kan bijvoorbeeld de meest waarschijnlijke referent van een pronomen vinden. De twee belangrijkste beperkingen van de Gestalt modellen zijn dat ze geen leestijden kunnen voorspellen en dat de representaties die ze ontwikkelen alleen handig zijn voor de taak die ze leren uit te voeren. Voor het Story Gestalt model betekent dit dat het niets weet over de volgorde van gebeurtenissen in een verhaal.

Het model van Golden en Rumelhart (1993) introduceert het idee van een situatieruimte. Punten in deze ruimte representeren situaties die kunnen voorkomen in een verhaal. Elke dimensie van de ruimte komt overeen met één propositie, en elke situatie bestaat uit een conjunctie van deze proposities. Een verhaal is een temporele reeks van situaties, wat wordt gerepresenteerd als een traject door de situatieruimte. Het model krijgt een dergelijk traject als invoer en transformeert het in een informatiever traject, waarbij gebruik wordt gemaakt van kennis over de oorzakelijke relaties tussen de proposities die de dimensies vormen van de situatieruimte. De wiskundige basis van het model wordt gevormd door Markov random field theorie, wat garandeert dat het traject dat het model vindt altijd het meest waarschijnlijke is, gegeven de beperkingen die het verhaal oplegt.

Het model van Golden en Rumelhart kampt met enkele tekortkomingen wat betreft de verhalen en de kennis die geïmplementeerd kunnen worden. Ten eerste kan het geen kennis implementeren over de relaties tussen proposities binnen één situatie van een verhaal. Ten tweede kunnen alleen verhaalsituaties worden gerepresenteerd die bestaan uit één propositie of uit een conjunctie van proposities. Ten derde kan het model geen kennis implementeren over de oorzaak of het gevolg van een conjunctie, als deze kwalitatief anders zijn dan de oorzaak of het gevolg van de individuele proposities die samen de conjunctie vormen. Er zijn redenen om te geloven dat al deze beperkingen het gevolg zijn van enkele basisaannamen over de architectuur van het model, in het bijzonder de aanname dat er geen invloed is tussen proposities binnen één moment in het verhaal. Deze aanname kan echter niet worden genegeerd zonder dat de Markov random field analyse ondoenlijk wordt.

De analyse van deze acht modellen maakt duidelijk dat het modellen van tekstbegrip vaak ontbreekt aan computationele degelijkheid. Zonder wiskundige grondigheid zijn zogenaamd computationele modellen niets meer dan verbale modellen vermomd in formules. Alleen de Gestalt modellen en het model van Golden en Rumelhart kunnen computationeel degelijk genoemd worden. Ook als we vier van Jacobs en Grainger's (1994) evaluatiecriteria voor modellen in aanmerking nemen, blijkt de vooruitgang op het gebied van tekstbegripmodellen teleurstellend. Om mee te beginnen worden veel modellen niet serieus gevalideerd aan empirische gegevens. Als modellen wel veel resultaten lijken te voorspellen is dit vaak niet hun eigen verdienste (het model van Langston en Trabasso, bijvoorbeeld) of doen ze dit alleen vanwege subjectief en ad hoc ingestelde parameterwaarden (het Construction-Integration model). Zelfs modellen die wel aan empirische gegevens worden gevalideerd verklaren vaak weinig (het Landscape model) of zelfs helemaal niets (het model van Langston en Trabasso). Andere modellen vereisen veel ad hoc formules en parameterwaarden (het Construction-Integration model) terwijl sommige niet eenvoudig kunnen worden toegepast op verschillende verhalen (het model van Golden en Rumelhart) of taken (de Gestalt modellen).

Het Distributed Situation Space model

In Hoofdstuk 4 wordt een model ontwikkeld voor kennisgebaseerde inferenties tijdens verhaalbegrip, dat niet alleen computationeel degelijk is, maar ook empirische gegevens voorspelt en verklaart, maar een klein aantal aannames vereist, verschillende verhalen kan verwerken, en verschillende taken kan uitvoeren. Dit Distributed Situation Space (DSS) model deelt een groot deel van zijn architectuur met het model van Golden en Rumelhart, wat computationele degelijk garandeert. Het belangrijkste verschil met Golden en Rumelharts model ligt in de representatie van verhaalsituaties. In de situatieruimte van het DSS-model is er geen één-op-één relatie tussen dimensies en proposities. In plaats daarvan is de representatie van een propositie verdeeld over meerdere dimensies, en codeert elke dimensie voor verschillende proposities.

De situatieruimte wordt ontwikkeld door informatie te onttrekken uit een handgemaakte beschrijving van gebeurtenissen in een microwereld. In deze microwereld leven twee personages, die in verschillende toestanden kunnen verkeren en verschillende activiteiten kunnen ontplooien. Een Self-Organizing Map (Kohonen, 1995) ontdekt regelmatigheden binnen situaties in de microwereld, en leert proposities representeren op een manier die overeenkomt met deze regelmatigheden. Er zijn drie voordelen aan deze representatie. Ten eerste kunnen niet alleen conjuncties, maar ook disjuncties en negaties van proposities (en daarmee elke situatie in de microwereld) worden gerepresenteerd als een vector in de situatieruimte. Ten tweede kan de a priori subjectieve waarschijnlijkheid op het vóórkomen van een situatie worden berekend uit zijn vectorrepresentatie. Ten derde volgen uit de vectorrepresentaties ook conditionele subjectieve waarschijnlijkheden, gegeven bepaalde proposities of situaties. De subjectieve waarschijnlijkheid van een propositie wordt zijn geloofswaarde genoemd, omdat het aangeeft in hoeverre een lezer zou kunnen geloven dat die propositie het geval is in een verhaal. Geloofswaarden worden ook gebruikt om maten te definiëren van propositiepassendheid en verhaalcoherentie, die gebruikt worden om de resultaten van het model te vergelijken met empirische gegevens.

Net als het model van Golden en Rumelhart, representeert het DSS-model een verhaal als een temporele reeks situaties, die een traject vormt door de situatieruimte. Kennis over temporele verbanden tussen opeenvolgende situaties in de microwereld wordt gebruikt om een dergelijk traject te transformeren in een waarschijnlijker traject, rekening houdend met de beperkingen opgelegd door de verhaaltekst. Uit de wiskundige basis van het model, Markov random field theorie, volgt precies hoe wereldkennis over temporele verbanden tussen situaties kan worden geïmplementeerd, en hoe een verhaaltraject beter met deze temporele wereldkennis in overeenstemming kan worden gebracht tijdens het inferentieproces van het model. Intussen kunnen geloofswaarden van proposities worden verkregen om vast te stellen in hoeverre deze proposities afgeleid zijn. Het punt waarop het inferentieproces stopt hangt af van een parameter die de verwerkingsdiepte regelt.

Uit de resultaten blijkt dat proposities inderdaad worden afgeleid als ze waarschijnlijk het geval zijn in het verhaal, gegeven de regelmatigheden in de microwereld. Verder voorspelt het model een redelijke hoeveelheid empirische gegevens. In overeenstemming met de empirie simuleert het model hoe het verwerken van een verhaalsituatie leidt tot meer inferenties en langere verwerkingstijd als de situatie zwakker verbonden is met voorgaande situaties. Ook het verhogen van de waarde van de verwerkingsdiepteparameter leidt tot toename in hoeveelheid inferenties en verwerkingstijd, wat overeenkomt met empirische bevindingen. Omdat bij het ontwikkelen van het model nooit ontwerpbeslissingen zijn genomen met bepaalde empirische gegevens in gedachten, is het feit dat de modelresultaten overeenkomen met empirische gegevens een emergente eigenschap. Er kan dus gezegd worden dat het model de empirische gegevens niet alleen voorspelt, maar ook verklaart waaruit deze gegevens voortkomen.

Het DSS-model is ontworpen om een gegeven verhaal om te zetten in het traject dat het waarschijnlijkst is volgens de wereldkennis van het model, met een nauwkeurigheid die wordt geregeld door een verwerkingsdiepteparameter. Dit is in overeenstemming met de hypothese dat het al dan niet maken van een inferentie niet direct afhangt van het soort inferentie, maar van de beschikbaarheid van relevante kennis, de mate waarin de inferentie bijdraagt de coherentie van het verhaal, en het doel van de lezer. Noordman, Vonk, en Kempff (1992) en Vonk en Noordman (1990) leveren experimenteel bewijs dat deze opvatting ondersteunt.

Het model heeft als emergente eigenschap dat het inferentieproces in het algemeen leidt tot verhoogde coherentie, zonder dat coherentie invloed heeft op het inferentieproces. Inferenties leiden tot een coherentere verhaalrepresentatie op een situationeel niveau. Dit is voor een groot deel in overeenstemming met de constructionistische theorie van inferentie (Graesser, Singer, & Trabasso, 1994) en niet met de minimalistische theorie (McKoon & Ratcliff, 1992), die beweert dat uitgebreide situationele representaties gewoonlijk helemaal niet gemaakt worden. Echter, volgens de constructionistische theorie zijn inferenties het resultaat van het zoeken naar coherentie, terwijl deze rollen zijn omgewisseld in het DSS-model, dat coherentietoename ziet als bijproduct van het infereren.

Uitbreiden van het DSS-model

Drie uitbreidingen op het DSS-model worden beschreven in Hoofdstuk 5. Ten eerste wordt aangetoond dat het model op een eenvoudige manier kan simuleren hoe verhalen in de loop van de tijd worden vergeten. Het traject dat voortkomt uit het inferentieproces kan worden gezien als het geheugenspoor dat door het verhaal is achtergelaten. Gedurende de tijd dat het verhaal onthouden wordt, wordt dit traject zwakker, wat wil zeggen dat de hoeveelheid informatie in het traject afneemt. Door aan te nemen dat delen van het traject langzamer hun informatie verliezen als ze sterker aan elkaar verbonden zijn, kan het onthoudmodel nabootsen hoe proposities die het minst bijdragen aan de verhaalcoherentie als eerste worden vergeten. Ook wordt correct voorspeld dat er minder wordt herinnerd naarmate de onthoudtijd toeneemt, dat proposities beter worden onthouden als ze beter in het verhaal passen, dat intrusie (dat wil zeggen, onterechte herinnering) waarschijnlijker is voor proposities die beter in het verhaal passen, en dat deze laatste twee effecten sterker worden naarmate de onthoudtijd toeneemt. Al deze effecten zijn ook empirisch gevonden.

Ten tweede wordt het DSS-model uitgebreid met een proces dat het oplossen van dubbelzinnige pronomina nabootst. Een bewering die een dubbelzinnig pronomen bevat, verandert in een microwereldpropositie wanneer een entiteit uit de tekst wordt gekozen als referent van het pronomen. Net zoals een normale propositie, kan de dubbelzinnige bewering kan worden gerepresenteerd als een vector in de situatieruimte. Deze vector is een combinatie van alle vectoren die een mogelijke uitkomst representeren van het pronomenoplossingsproces. Deze combinatie wordt aangepast aan de mate waarin verschillende entiteiten in focus zijn, op zo een manier dat de vector voor de dubbelzinnige bewering dichter in de buurt komt van de vector die het resultaat is van het kiezen van de sterkst gefocuste entiteit als referent van het pronomen.

Een keuze tussen mogelijke referenten wordt geforceerd door de vector voor de ambigue bewering te laten 'vallen' naar de gebieden in de situatieruimte die overeenkomen met gedesambigueerde beweringen. Dit proces wordt beïnvloed door het inferentieproces van het DSS-model, waarmee het effect van contextinformatie op het oplossen van een pronomen wordt geïmplementeerd. Het pronomenoplossingsmodel simuleert hoe de eerste interpretatie van een pronomen afhangt van de focus, en hoe die teniet kan worden gedaan door contextinformatie die met de focus in tegenspraak is. Verder voorspelt het model empirische gegevens betreffende leestijden en foutpercentages, en verklaart het hoe deze worden beïnvloed door focus, contextinformatie, en verwerkingsdiepte.

De derde uitbreiding op het DSS-model is een meer tekstueel niveau van representatie. Een neuraal netwerk met terugkoppeling wordt getraind om zinnen, die situaties beschrijven, om te zetten in de DSS-vectoren die deze situaties representeren. Voor deze taak is een microtaal ontwikkeld waarin microwereldsituaties worden beschreven. Het netwerk leert redelijk goed om de zinnen te verwerken. De tussenliggende representatie die daaruit ontstaat blijkt tekstuele, propositionele, en situationele aspecten van een tekst te combineren in een enkel representatieniveau.

Conclusies

Het DSS-model en zijn uitbreidingen schetsen een beeld van tekstbegrip dat sterk afwijkt van de opvatting die heerst sinds het invloedrijke artikel van Kintsch en Van Dijk (1978). Net zoals Golden en Rumelhart (1993) hebben wij het situatiemodel en zijn relatie tot de kennis van de lezer centraal gezet, in plaats van ons te richten op tekstproposities en hun relatie tot de tekst. Het DSS-model gebruikt geen propositionele structuren maar beschouwt proposities als feiten over een microwereld, en verklaart hoe wereldkennis de interpretatie van binnenkomende feiten beïnvloedt. In onze opvatting komt het begrijpen van een tekst neer op het vormen van een situationele representatie. Lagere representatieniveaus bestaan alleen omdat ze nuttig zijn voor het omzetten van een tekst in een situatiemodel. Hoewel de betekeniseenheden waarin een tekst wordt gerepresenteerd op één van deze niveaus propositieachtig kan zijn, is er geen behoefte aan propositionele predikaat-argument structuren.

De nadruk die het model legt op kennis en hoger-niveau mentale representaties brengt twee bezwaren met zich mee. Ten eerste is elk model dat zich met kennis bezighoudt noodzakelijk beperkt tot een microwereld, zolang voorkomen moet worden dat kennis subjectief en ad hoc wordt toegepast. Het probleem met een microwereld ligt uiteraard in zijn beperkte afmetingen. Het is nog een open vraag in hoeverre het model met realistischer hoeveelheden kennis kan omgaan. Omdat het model direct de beschikking heeft over wereldkennis in de vorm van de vectorrepresentaties van proposities, hoeft het zich niet door een kennisbestand heen te worstelen. Er lijkt dus niet bij voorbaat een reden te zijn om te verwachten dat het model niet langer werkt wanneer veel meer kennis wordt toegevoegd. Het implementeren van deze kennis zou echter problematisch kunnen zijn vanwege technische beperkingen. Het trainen van een erg grote Self-Organizing Map kan te bewerkelijk blijken, terwijl een kleinere tot onnauwkeurige implementatie leidt.

Het tweede bezwaar betreft het gebrek aan realistische invoer. De series feiten die het model verwerkt, worden op een situationeel niveau gerepresenteerd, terwijl de tekst waaruit deze feiten afkomstig zijn, wordt genegeerd. De invoer zou eigenlijk evengoed op iets anders dan een tekst gebaseerd kunnen zijn, een film bijvoorbeeld. Voor het model maakt dit helemaal niets uit. Welke vorm de realistische invoer ook aanneemt, hij moet worden omgezet in de situationele vectorrepresentatie. In Sectie 5.3 is aangetoond dat een neuraal netwerk met terugkoppeling, getraind kan worden om series woorden om te zetten in de bijbehorende situatievectoren. Een tekst bevat echter veel aanvullende, tekstuele informatie die het begripsproces kan beïnvloeden maar niet kan worden verwerkt door het huidige model. Een voorbeeld van zo een bron van tekstuele informatie is het pronomen. Er is aangetoond dat het DSS-model eenvoudig kan worden uitgebreid om het oplossen van pronomina na te bootsen, maar dit is slechts een kleine, eerste stap in het veranderen van het verhaalbegripsmodel in een tekstbegripsmodel.

Nawoord

Ὥρη μὲν πολέων μύθων, ὥρη δὲ καὶ ὕπνου.[1]

Na zes hoofdstukken, twee appendices, en een samenvatting over het begrijpen van verhalen, wil ik tenslotte graag wat verhalen *vertellen*. Verhalen over iedereen aan wie te danken is dat ik nu dit Nawoord kan schrijven. Dat begint natuurlijk met Wietske, Leo, en Mathieu, want zonder hun onuitputtelijke hulp was het nooit gelukt. Ik kan me niet voorstellen dat er ooit, ergens, iemand betere begeleiding heeft gekregen. Uiteraard gaat mijn dank ook uit naar de rest van de manuscriptcommissie: Walter Daelemans, Gerard Kempen, Antal van den Bosch, en vooral Rein Cozijn.

Zonder Sophia waren de voorgaande 220 pagina's er ook nooit gekomen. Als zij me er niet van had overtuigd dat Nijmegen helemaal niet ver weg is, was ik er nooit aan begonnen. En als zij me niet telkens weer had doen inzien dat het allemaal wel goed zou komen, dan had ik het al tien keer opgegeven.

Ook al bleek Nijmegen dus niet zo ver als ik vroeger dacht, toch had ik het treinreizen nooit volgehouden als het daar niet zo gezellig was. Dat was natuurlijk te danken aan alle collega's bij het IWTS, in het bijzonder Pim en Mark vanwege het tafeltennissen, Mirjam voor het opeten van mijn chocoladeoverschot, Femke omdat ze me de beste bioscoop van Nederland heeft laten zien, en Roel, die me naar de beste café's van Nijmegen heeft meegenomen (alleen over The Shuffle had hij me nooit iets verteld) en het aandurfde mijn paranimf zijn. Net als Femke had hij ook altijd een matras voor me beschikbaar wanneer dat nodig was, en ik hoop het nog vaak nodig te hebben. Natuurlijk wil ik ook mijn vrienden in Amsterdam en Leiden niet overslaan. Vooral niet Thessa, die zich zelfs door haar emigratie niet laat weerhouden van het paranimfschap.

Tenslotte wil ik graag mijn ouders en Erika bedanken voor het niet doorlopend vragen hoe het nou met m'n proefschrift staat. Dat heeft geholpen.

Goed, de tijd voor verhalen zit er eindelijk op. Het is tijd voor slaap.

Amsterdam, december 2003

[1] Er is een tijd voor vele verhalen, er is ook een tijd voor slaap (Homerus, *Odyssea*).

Curriculum Vitae

Stefan Frank was born in Sassenheim, the Netherlands, on June 11th, 1973. He studied Chemistry at Leiden University for one year before determining that he would prefer to study Artificial Intelligence at the Free University of Amsterdam. During this period, he worked as a teaching assistant for a Prolog-programming course. After obtaining his M.Sc. degree in April 1998, he worked as assistant-lecturer for a few months, preparing a course in data mining. In 1999, he started the research project 'a model for text comprehension', funded by the Netherlands Organization for Scientific Research (NWO), stationed at Tilburg University but working from within the Interfaculty Research Unit for Language and Speech (IWTS) of the University Nijmegen. He finished this thesis in November 2003, and is currently enjoying a short-term post doctoral position at the Nijmegen Institute for Cognition and Information (NICI).