Report #2

# A Computational Model of On-line Story Understanding

Six Month Progress Report: 21 April 1997

Student: Elliot Smith

Supervisor: Peter Hancox

Other members of thesis group:

Ela Claridge

Mark Torrance

## Abstract

Models of story understanding in artificial intelligence and cognitive psychology typically concentrate on the construction of representations of story content. Construction of a ‘complete’ representation (informally equivalent to an ‘understanding’) requires inferences which establish coherence links between text statements. This report examines how *reality-based* links (e.g. causal connections) and *narrative-based* links (e.g. those that establish rhetorical structure and delineate text segments) have been handled in previous models. My suggestion is that both types of links are required for narrative understanding, and that each is dependent on the other. I also indicate how the inferences required to create both types of link may be uniformly represented and processed within a story grammar framework. This framework could also be used to model theories about recall (hierarchical storage) and processing.

## Contents

# 1. Introduction

## *1.1. Why Story Understanding?*

Story understanding is a classic problem in Artificial Intelligence (AI) and is interesting from many perspectives:

- *Search, control and representation*
  Story understanding involves three important aspects of AI: search, control, and representation. My own interest arose from looking at the inferencing process during story understanding, which corresponds to a search problem. The control of search is a problem which occurs in many branches of AI, such as planning, problem solving, and theorem proving. Scripts are one means of implementing (inferential) control, and are specifically geared towards story understanding (see section 2).

- *Cognitive modelling*
  Computer-based and theoretical models of how *people* understand stories and texts have been developed in cognitive psychology (e.g. Thibadeau et al., 1982; Kieras, 1982). These models are directly concerned with forming hypotheses to explain experimental data.

  My main interest in these simulations is their attempt to account for story analysis, given a *resource-bounded* system. The limitations of memory imply that there must be a mechanism which can prevent a combinatorial explosion of inferences (the bane of many AI systems). These models attempt to show how a representation of a story can be incrementally derived, given a limited-capacity working memory.

  Some AI systems have similar goals, as processing time and computer memory limitations bound the operation of any program. However, many (story) understanding systems place no limits on the length of inference chaining (e.g. Hobbs et al., 1993).

- *Engineering solutions*
  I consider story understanding to be a subset of the field of text understanding. From an engineering point of view, text understanding is a profitable area, as it can produce practical programs, e.g. for extracting information from texts (Mikheev, 1996), or answering questions about them (Lehnert et al., 1983). Descriptions of implemented systems are a useful resource, as they describe how to deal with the reality of computer processing.

In the next sections, I describe models that have been developed in AI and cognitive psychology, along the lines of the schema-based paradigm. As I attempt to show, the emphasis in the two fields differs. AI models are concerned primarily with low-level processing issues (e.g. parsing, inferential efficiency); cognitive psychology models are concerned primarily with explaining experimental data (e.g. recall, working memory limitations).

### *1.2. Schemas*

Bartlett (1932) is generally credited with founding the psychological and computational study of story understanding. Bartlett examined how appropriate schemas were used during the recall of stories. A schema is ‘a hierarchically organised set of units describing generalised knowledge’ (Mandler, 1984). This knowledge may be about the structure of a sequence of events, a scene, a plan, a goal, or any other ‘thing’ that can be described as a set of smaller sub-components. Much subsequent work in artificial intelligence and cognitive psychology has used the schema-based approach as a basis for work in various areas, including planning, vision, and language understanding.

### *1.3. Schema-Based Approaches in Artificial Intelligence*

Minsky’s frames (1975) are an early example of an AI model explicitly based on the idea of a schema. Minsky devised frames as a psychologically valid data structure for storing information about ‘things’ in the world (specifically, visual information). A frame stores default information about the attributes of a ‘thing’ (e.g. an object, a room, or a situation) as a hierarchical ‘network of nodes and relations’ (Minsky, 1975). The nodes of a frame are called *slots*. Slots contain default information that can normally be inferred when faced with a new instance of a particular class of ‘things’. In this sense, slots supply *expectations* about the properties of members of a class. However, if a thing varies from the default values supplied, the default can be overridden and the actual value used instead. A classic example of this is the frame for bird: we may infer, by default, that any bird we encounter can fly. However, if we find that a particular bird is a penguin, and therefore cannot fly, this information overrides the default value in the ‘can-fly’ slot.

Schank and Abelson formalised a specialised version of the frame, the *script*, designed specifically for the purpose of story understanding [1]. Scripts, or variants of them, were used in most of the story understanding systems developed at Yale in the late 1970s/early 1980s (Lehnert’s FRUMP; Cullingford’s SAM; Dyer’s BORIS; Wilensky’s PAM). Unlike frames, scripts also incorporate information about temporal sequences and processes (e.g. people order a meal before paying for it) [2]. A script for a sequence of events is comparable to a schema for that sequence: it consists of a set of expectations about the way in which that sequence proceeds (Mandler, 1984).

The strength of scripts is their utility as models of story processing. They are discussed in more detail in section 2.

---

[1] Earlier work on scripts was carried out by Charniak (1972) and Schank (1975), but Schank and Abelson’s (1977) book was the first full-fledged attempt at formalisation of these ideas.

[2] This distinction between *static* information and *sequential* information has been made by several people: e.g. Bartsch’s (1987) analytic frames (static) and scenic frames (sequential); de Beaugrande and Dressler’s (1981) distinction between frames (static) and scripts (sequential); and Schank’s MOPs (Dyer, 1992), which encode both sequential and static information. This distinction is not unproblematic, e.g. when does a state become a process? can frames contain temporal information?. However, it does highlight the need for a representation that can handle both static and sequential knowledge.

### *1.4. Schema-Based Approaches in Cognitive Psychology*

While research on scripts and AI approaches progressed, cognitive psychologists also became interested in the structures and processes underlying story comprehension. One outcome of this was the story grammar approach, which was very popular at the end of the 1970s.

Story grammars originated from the work of Propp (1928). Propp was a structuralist literary critic, whose work focused on the structure of folktales. He claimed that folktales from the same culture shared certain formal properties. Lakoff (1972) introduced story grammars into psychology.

Story schemas and story grammars differ in that a schema is a mental construct, whereas a story grammar describes how story statements can be inserted into a schema. The terms ‘schema’ and ‘grammar’ are used differently by different authors, and are often confused with one another. One of my aims is to clarify and formalise the difference between story grammars and story schemas.

Also note that I use the phrase ‘story statement’ throughout this paper to refer to a clause or sentence that expresses a single event or state that can be represented as a predicate/argument expression (after Mandler, 1984). I haven’t attempted a strict formalisation of ‘story statement’; however, using it prevents the confusion caused by using the phrase ‘proposition’, which is often used to denote a ‘story statement’ in psychological studies of story understanding (e.g. Rumelhart, 1975). The psychological use of ‘proposition’ is more casual than its use in classical logic. (For example, in classical logic, propositions have truth values, which are often of little consequence to psychologists.)

The strength of story grammars is their utility as models of story memory. Story grammars are discussed in more detail in section 3.

## 2. Scripts as Models of Story Processing

### *2.1. Introduction*

The script-based approach provides a practical processing model for stories. Unlike psychological models, which often remain theoretical, scripts have been implemented in computer programs. This has meant that one of the central problems of story understanding has been addressed in script-based research: the problem of inferencing.

### *2.2. Scripts and Inferences*

A good working definition of an inference is provided by Norvig (1989):

> ...any assertion which the reader comes to believe to be true as a result of reading the text, but which was not previously believed by the reader, and was not stated explicitly in the text.

Inferences are central to story understanding, as they bridge gaps in the reader's understanding of a story, and create meaningful relationships between story statements. The information used to bridge these gaps is *implicit* in the story. Any text, including a story, contains both *implicit* and *explicit* information (Kintsch and van Dijk, 1978). Implicit information is recovered by inference making, from the context provided by the story, general world knowledge, and knowledge about narrative conventions.

For example, an inference is needed to make the following paragraph coherent (Black and Wilensky, 1979):

> Margie was tightly holding the string of her balloon. Suddenly, a gust of wind caught it and carried it into a tree. It hit a branch and burst. [Margie was sad.] Margie cried and cried.

The section of this example in square brackets indicates that the information was supplied by an inference. The inference 'Margie was sad' is derived from general world knowledge, rather than from the text itself: when people lose a treasured object, they are upset.

## 2.3. Inferential Promiscuity

Note that there are other inferences that could have been made while reading the story above, for example:

- Margie's balloon was red.
- The tree was an oak.
- Margie was ten years old.
- Margie stubbed her toe, which made her cry.
- Margie had a father called Fred.

The difference between the inference 'Margie was sad' and these inferences is that 'Margie was sad' is a *proper* inference: it is *plausible*, *relevant*, and *easy* (Norvig, 1989). The inferences given above are *promiscuous*: they are arbitrary, irrelevant, and convoluted, and do little to improve the informativity and coherence of the text.

The inference 'Margie was sad' is a proper inference as it provides an *explanation* for her tears, and is *caused* by her bursting her balloon. The inference creates coherence that would otherwise not exist between the two sentences, and adds information to the reader's understanding. This suggests that only *proper* inferences should be maintained and incorporated into the representation of the text, while promiscuous inferences should be controlled and/or rejected.

### 2.4. How Scripts Help

Scripts provide a means of controlling the inferences made during text comprehension, by providing expectations about what will happen next in a text (Dyer, 1981). Expectations are represented as slots within a script; the processor looks for fillers for these slots during reading. A slot may be filled by a single statement in the story, or by a group of statements that comprise a sub-frame or sub-script. For example, a $HOLIDAY script may have a slot called &JOURNEY_OUT, which could be filled by a group of statements describing a car journey [3]. Conditions can also be attached to a slot, specifying which parts of a text can be used as a filler (for example, only humans can fill a &WAITER slot).

If a filler for a slot cannot be found in the text, a script *supplies* a plausible default for the slot. That is, it allows an immediate inference about the content of that slot. For example, if a story about a restaurant visit doesn't mention eating food, the script supplies the information that food was eaten.

Scripts also *constrain* the inferences that can be made during reading, preventing a combinatorial explosion of inferences. Only inferences which contribute to filling slots are allowed by the script, enforcing the requirement for inferential relevance.

### 2.5. Limitations of Scripts

Scripts look promising so far, but in practice there are several problems:

- *'World-centred' description of texts*
  Scripts tend to concentrate on event sequences, which are *reality-based*. That is, scripts ignore 'literary', descriptive information, and subordinate the story to a single 'mundane' causal chain (Wilensky, 1983). Story understanding also requires *narrative-based* structures, such as episodes and settings (Rimmon-Kenan, 1983). While such structures have no utility as part of an event sequence, they help explain expository text that occurs in a story. Scripts generally ignore this descriptive information.

- *Monostratal memory storage*
  Scripts have limited utility as hierarchical models of memory, unlike story grammars. Scripts (traditionally) store all the information about a story on a single level (they are *monostratal*). Everything that occurs during a story is remembered with the same degree of detail. By contrast, story grammars suggest that different parts of stories are subordinated to different sub-structures, which can be completed before the whole story has been processed. This is an important concept, as it means that only information relevant to the current section of the story has to be available in working memory.

---

[3] The notation used here is taken from Dyer (1992), with the '$' prefix denoting a script, and '&' denoting a slot.

### *2.6. Summary*

Scripts represent a story as an event sequence, stored as a series of slot fillers. The representation for a particular story is a sequence of slots containing appropriate fillers. Slots may be filled by single story statements, or by frames and/or scripts. Fillers can be supplied by the text, or by inferences allowed by the script(s).

Scripts are a useful model of story processing, as they provide some answers to the problems of inferential promiscuity. They do this by supplying and constraining the inferences that can be made during processing.

Scripts face problems when presented with information that does not directly contribute to filling an ‘action’ or ‘event’ slot. This ignores the significance of higher-level narrative structures, to which descriptive text may make a contribution. In addition, scripts make no predictions about text recall: script-based systems remember and process information flawlessly. They are also informal and ill-disciplined, relying on ad hoc solutions to representation and processing issues.

## 3. Story Grammars as Models of Story Memory

### *3.1. Introduction*

Unlike scripts, which have been directly implemented in computer programs, story grammars have remained largely theoretical (although some implementations are discussed in section 4.1). As a result, processing models for story grammars are relatively underdeveloped.

However, an explicitly grammatical approach has several advantages:

- Story grammars provide a means for describing ‘interesting’, ‘literary’ stories, rather than just describing event sequences. They can do this by representing story-level constituents, such as episodes and settings, as well as event-level constituents (see section 3.3.1).

- Story grammars have been extensively tested and show some psychological validity, such as explaining the *levels effect* during recall (see section 3.3.2).

- Story grammars suggest the possibility of integrating a grammar for sentences with a grammar for higher-level constituents. In addition, the notion of parsing as an inferencing process (Shieber et al., 1995) parallels the idea of understanding as an inferencing process (presented most strongly by Hobbs et al., 1993). Perhaps the parsing and understanding processes could also be integrated with each other? These observations also suggest that current techniques in other areas, such as Discourse Representation Theory and modern grammar formalisms, could be used to replace the ad-hoc stylings of scripts, allowing formalisation of grammar, parsing, inferencing, and representation of script-like structures. I briefly describe these ideas in section 4.

### *3.2. The Structure of Story Grammars*

A traditional phrase structure grammar rule might be of the form:

```
Sentence --> NounPhrase, VerbPhrase
```

This means 'A `Sentence` comprises a `NounPhrase` followed by a `VerbPhrase`'. Symbols on the right hand side of a rule that can be further expanded by other symbols are called *non-terminals*. Symbols on the right hand side that correspond directly to strings in the language are called *terminals* (Rich and Knight, 1991). The parse tree derived from a sentence describes the relationships between its constituents in terms of a *dominance hierarchy*. Another way of saying 'A `Sentence` comprises a `NounPhrase` followed by a `VerbPhrase`' is 'A `Sentence` dominates a `NounPhrase` followed by a `VerbPhrase`'.

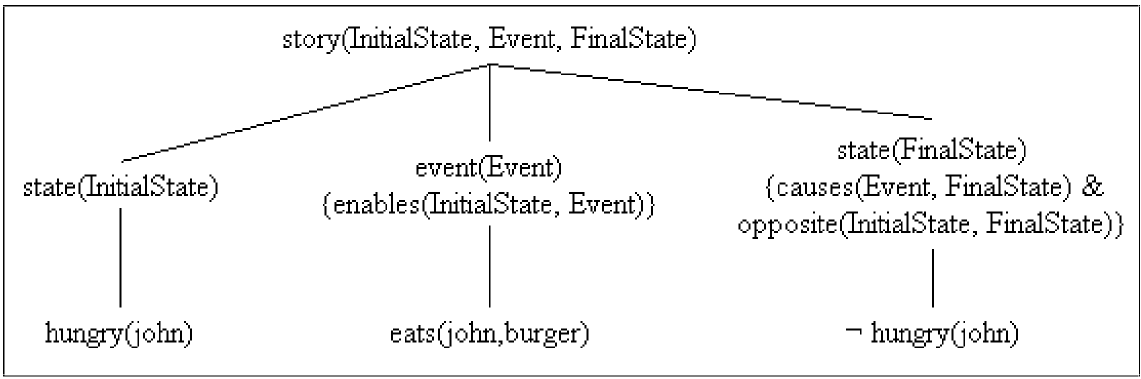
Story grammars take advantage of this notation to describe the structure of stories. The *terminals* of a story grammar are equivalent to story statements. Story statements are dominated by non-terminal categories such as *events* and *states*, which are themselves dominated by higher level categories, such as *episodes* and *settings*. The top node of a *story tree* will always be the *story* node (Correira, 1980). A grammar rule describing a set of possible 'toy' stories (without episodes, settings and so on) might look like this:

```
story(InitialState, Event, FinalState) -->
         state(InitialState), event(Event), state(FinalState),
         {allows(InitialState, Event) & causes(Event, FinalState)
         & opposite(InitialState, FinalState)}.
```

The notation I have used is an ad hoc one, based on Rumelhart (1975) but modified for perspicuity. In keeping with standard descriptions of story grammars, I've used story statements as the terminal nodes, bypassing the question of how the actual sentences of the text become story statements.

The rules between the '`{`' and '`}`' are conditions that must hold between the constituents. For example, the condition `causes(Event, FinalState)`specifies that `FinalState` must in some sense be a direct result of `Event`. To use a more formal definition of this relationship, we could say that `Event` *entails* `FinalState`. I'm not going to go into any great detail about how such relationships can be determined, but will assume that there is some way of determining them.

An example tree for the story 'John is hungry. John eats a burger. John isn't hungry as a result.' is shown below, based on the grammar rule given above.



### *3.3. Evidence for Story Grammars*

An obvious question might be whether story grammars have any psychological validity. There have been many experiments which demonstrate evidence for the predictions made by story grammars. I'll briefly discuss two areas of interest:

- The validity of story-level constituents.
- The levels effect in recall, which can be explained by reference to story trees.

#### *3.3.1. The Validity of Story-Level Constituents*

Black and Bower (1979) have demonstrated the validity of *episode* constituents. An episode is generally defined as a sequence of events and states that leads from a problem condition to a conclusion (Mandler and Johnson, 1977). While an event or state is often considered to be an *atomic* or *primitive* unit of a story, episodes are higher level constituents that contain one or more embedded atomic units.

Black and Bower showed that reading rate was affected by episode constituents: at the boundaries of episodes within a story, reading rate slows down. Presumably, this has to do with the fact that when an episode ends (for example, a problem is resolved by the protagonist), the episode concerned is 'wrapped up' (Thibadeau et al., 1982); that is, the constituent is completed and integrated into the story tree constructed so far.

#### *3.3.2. The Levels Effect in Recall*

A story tree is inherently *hierarchical* in terms of the information it contains (Thorndyke, 1977): the higher a node is in the story tree, the more abstract the information it contains. In addition, nodes higher up the story tree will be recalled more frequently and with greater accuracy than nodes lower down the tree. In other words, a node which dominates one or more other nodes has a greater chance of being recalled than the nodes it dominates, as it contains the 'gist' of the nodes below it (Kintsch and van Dijk, 1978). This is called the *levels effect*.

For example, if we have the paragraph

> Jack goes to a restaurant. He orders a burger. He eats it, then pays and leaves.

we could sum up its gist by placing it under a 'restaurant_visit' node. Information about who did what where could annotate this node, for example 'restaurant_visit(Jack, burger)'. (Note that a node presented in this way looks like a script: see section 4.1 for further discussion.)

### *3.4. Similarities Between Scripts and Story Grammars*

Several researchers have criticised the idea that stories can be described by grammatical structures in the same way that sentences can (e.g. Wilensky, 1983; Black and Bower, 1980). Often, they suggest alternative approaches, such as using plot units, points, causal chains, or inferring character goals and intentions. However, I feel that the distinction being made is in many cases a false one: these ‘alternative’ methods themselves rely on stored structures that describe *sequences* of events, propositions, story statements, etc..

Most story understanding systems that are based on stored structures (of whatever granularity) are, in this sense, ‘grammatical’. For example, in the previous section, I showed how a dominating node within a story tree could be considered to be a kind of script. Schank and Abelson (1977) themselves suggest something similar, when they describe how a whole causal chain could be reduced in memory to its main conceptualisation, annotated with the appropriate slot fillers. Such *macroscopic events* (macroevents) are equivalent to a sequence of events, and represent the ‘gist’ of the sequence. Several such macroevents could be chained together to represent even higher level structures: for example, we could add ‘cinema visit’ and ‘drink in the pub’ macroevents, and subordinate these two and the ‘restaurant visit’ macroevent to an even larger ‘night out’ macroevent [4].

The implication is that script-like sequences can be described within the context of a story grammar. I return to this topic again in section 4.1.

### 3.5. Problems with Story Grammars

Two of the most frequent criticisms of story grammars are:

- *Story grammars are formally inadequate*
  Black and Wilensky (1979) criticise story grammars for their inability to recognise (in the formal grammatical sense) all possible stories. Some stories don’t have the canonical form that story grammars imply: constituents might be moved around for effect, e.g. in a murder mystery story, or a story involving flashbacks. In addition, constituents may be deleted, requiring inferences to fill the gaps left behind. As a result, story grammars are formally inadequate as they fail to define transformations which can account for the movement and deletion of constituents. There are two ways to reply to this criticism.

  Firstly, it may be that transformations of the kind envisaged by Black and Wilensy are not necessary in sentence-level grammars. Some seemingly context-sensitive phenomena, such as unbounded dependencies, can be handled with various *meta-rules* or similar mechanisms [5]. Recent approaches to grammar, such as dependency grammar (Covington, 1990), generalised phrase structure grammar (Gazdar et al. 1985), and combinatory categorial grammar (Ades and Steedman, 1982), incorporate such mechanisms for dealing with sentence level phenomena, without recourse to transformations.

---

[4] Memory Organisation Packages (MOPs) were developed to capture these possibilities. MOPs are hierarchically organised, and are finer grained than scripts. New, larger scripts can be formed by agglomerating individual MOPs (Dyer, 1981).

[5] Gazdar and Mellish (1989) state that even finite state languages ‘can exhibit dependencies between symbols arbitrarily far apart.’

Secondly, the critcism on the basis of mathematical adequacy misdirects attention from the more important question of *notional adequacy*. As Rumelhart (1980) points out, ‘The psychologically interesting thing about a grammar is that it proposes an analysis of the *constituent structure* of a linguistic unit’. An analysis of ‘what elements “go together” to form higher elements and how one group of elements is related to another’ should be the primary focus of any story grammar, rather than whether the grammar describes all possible stories. This shifts the focus on to examining the categories and sequences proposed by story grammarians, to see whether they fit with evidence and intuitions about the structuring of stories.

- *Story grammars aren’t parsers*
  There is a confusion in the story grammar literature over how a grammar is used during processing. In many cases, the distinction between a grammar and a parser is not made clear.

  When you consider the problems of assigning categories to individual text segments, the problem of actual processing becomes more acute (Garnham, 1983). How are individual story statements assigned to the constituent categories of a story grammar? If there is to be a parsing mechanism analogous to a sentence parsing mechanism, there must be some procedure for assigning the lexical items of the story (i.e. the story statements derived from it) to constituent categories. For example, in a sentence grammar, the word ‘the’ is assigned to the category of *determiner*. A similar mechanism must exist in a story parser. For example, a story statement like ‘hungry(John)’ must be assigned to some kind of *state* category.

### 3.6. Summary

A story grammar is a useful model of story memory, as it describes the relationship between different levels of story understanding. A story grammar represents a story as a sequence of constituents, stored as a tree. Nodes of the tree may be individual events, or higher level structures like episodes.

Story grammars face problems when anyone attempts to implement them. Frequently, these problems are caused by researchers confusing grammars and parsers. Often, they also fail to describe how individual story statements can be assigned to grammatical categories, and how inferences fit into the picture.

## 4. Implications and Future Work

### 4.1. Representing Story Grammars Using Discourse Representation Theory

As I hinted in section 3.4, script-like sequences could be described within the context of a story grammar. This has also been suggested by Hobbs (1990), Correira (1980), and Bartsch (1987). With this in mind, it is worth considering how to design a representational system that can be used both to describe low level structures, like individual story statements, to high level structures, like scripts.

One system that offers a means of describing the content of a text is Discourse Representation Theory (DRT) (Kamp, 1981; Spencer-Smith, 1987). According to DRT, each text can be assigned a Discourse Representation Structure (DRS) that represents its meaning. DRSs can be created incrementally while a text is being read, by adding new information to the DRS as each sentence is encountered. Eventually, a single DRS for the whole text is created. DRT has been implemented in several Prolog programs at the University of Georgia, demonstrating its utility as a computational formalism (Covington et al., 1989).

DRT can be seen as a framework for uniform description of semantic structures. For example, Brown (1994) has shown how events can be represented by extending DRT. Bartsch (1987) goes further, showing how DRT can be used to describe objects ranging from scripts to case frames for verbs. She claims that the DRS for a whole text could be created compositionally during parsing, by merging the DRSs for lower level text constituents. For example, individual words would be assigned DRSs, which would be merged together to form the DRS for a clause or sentence. The sentence-level DRSs could also be merged together, eventually producing the DRS for the whole text.

Bartsch's theory boils down to the idea of *partitioning* the DRS for a text: the individual parts of the DRS are compositionally integrated together to form the DRS of the whole. DRT may be a formalism which could therefore be used to represent a story grammar: individual constituents described by a story grammar correspond to DRSs; the *story* node at the top of the tree corresponds to the DRS for the whole story.

## *4.2. Parsing with Story Grammars*

If a story grammar is to be practical, a parser will be needed to apply the grammar to a story. Very little research has looked at supplying parsers for grammars above the sentence level, but some research is briefly described below.

### *4.2.1. Control Strategies for Story Parsing*

As in sentence processing, there are different parsing control mechanisms that could be used to apply a story grammar to a given story. Two implemented story grammar parsers are described below.

*Bottom-Up Search*
Zhang Songmao has implemented a simple context-sensitive story grammar and parser. The system represents a story as a *case frame forest*. A *case frame* is a semantic tree representing a sentence, and roughly corresponds to a single story statement; a *case frame forest* is a set of case frames describing a whole story (Zhang Songmao, 1994). The parser matches case frames to the constituent categories of the grammar, eventually producing a story tree. The system presented can handle a few (very specific) stories.

There are two problems with this approach:

- The parser works bottom-up, without top-down guidance. Most researchers agree that both top-down and bottom-up processes are involved in story comprehension (e.g. Thibadeau et al., 1982).

- The story is pre-parsed into case trees, representing the story statements. There is no mechanism for converting the text of the story into case frames.

*Bottom-Up Search with Top-Down Guidance*
Correira's BUILDTALE system uses rules that are similar to Zhang Songmao's. However, he employs a mixed processing strategy, rather than simply a bottom-up one: narrative-based constituents are built top-down (e.g. episodes, settings), while reality-based constituents are built bottom-up (e.g. events and states) (see section 2.5 for more on this distinction). The *levels effect* is reproduced by reading off the nodes of the story tree at a particular depth.

This search strategy is comparable to that of a left-corner parser, which combines both top-down and bottom-up processing. This has obvious corollaries with the kinds of processes which occur during comprehension; it also arguably has a degree of psychological plausibility (Crocker, 1996).

### *4.2.2. Assigning Story Statements to Constituent Categories*

There is an 'assignment problem' in both story grammars and scripts: given a particular abstract representation of a fragment of text (e.g. a conceptual dependency or predicate logic expression), how can it be assigned to a node in a story tree, or a slot in a script?

In the case of story grammars, there are few attempts to describe a method of assigning categories to story statements. One method might be to express the rules of the story grammar as a case frame forest (Zhang Songmao, 1994 - see the previous section). This set of representations is then successively matched against the right-hand sides of rules, allowing the left-hand sides to be derived. (A similar technique is used in Correira, 1980.)

Another alternative would be to use *discourse relations*. It is clear that stories (and any kind of discourse) are not context-free, in the sense that individual story statements, such as states or events, at least partially rely on their 'satellites' for their meaning. Inferring relations between story statements and higher level constituents might provide a decision procedure for assigning story statements and other constituents to higher-level constituent categories, as outlined in Rumelhart (1975). These relations can be used to define how story constituents are connected to each other, and the kind of constraints that must hold between adjacent constituents for a story to be coherent.

Rumelhart provides no clear formal description of these relationships, but other researchers have looked at providing just such a description: the Linguistic Discourse Model (Polanyi, 1995), Rhetorical Structure Theory (Mann and Thompson, 1988), and Hobbs' coherence relations (Hobbs, 1979; Kehler, 1995) all provide formalised descriptions of relations that can hold between discourse constituents.

This approach suggests a locus for inference processes that occur during reading: inferences are made to satisfy conditions on nodes of the story tree. These inferences could be based both on world knowledge and knowledge about the usual structure of narratives. The conditions on the nodes could specify relationships that must hold between different parts of the story tree, how parts can be integrated with each other, which parts of the story are more important than others, where new statements can be attached to the tree, etc.

## 5. Timetable

| Period | Activity |
|---|---|
| April 1997 | Complete literature search (Computer and Control, LLBA) |
| May 1997 - June 1997 | Draft thesis proposal |
| July 1997 | Complete thesis proposal |
| August 1997 | Vacation |
| September 1997 - September 1998 | Initial coding and simultaneous write-up |
| October 1998 - March 1999 | Literature review and coding |
| April 1999 - October 1999 | Complete thesis |

## 6. Bibliography

Ades, A.E., Steedman, M.J. (1982) On the Order of Words. *Linguistics and Philosophy* 4, 517-558.

Bartsch, R. (1987) Frame Representations and Discourse Representations. *Theoretical Linguistics* 14, 65-117.

Black, J.B., Bower, G.H. (1979) Episodes as Chunks in Narrative Memory. *Journal of Verbal Learning and Verbal Behaviour* 20, 267-275.

Black, J.B., Wilensky, R. (1979) An Evaluation of Story Grammars. *Cognitive Science* 3, 213-230.

Black, J.B., Bower, G.H. (1980) Story Understanding as Problem Solving. *Poetics* 9, 223-250.

Brown, D.W. (1994) *A Natural Language Querying System Based on Discourse Representation Theory and Incorporating Event Semantics*. Artificial Intelligence Programs Research Report AI-1994-03, University of Georgia, Athens.

Charniak, E. (1972) *Toward a Model of Children's Story Comprehension*. MIT AI Lab Technical Report AI-TR-266, Cambridge, Mass.

Correira, A. (1980) Computing Story Trees. *American Journal of Computational Linguistics* 6(3-4), 135-149.

Covington, M.A., Schmitz, N. (1989) *An Implementation of Discourse Representation Theory*. Advanced Computational Methods Centre Technical Report, University of Georgia, Athens.

Covington, M.A. (1990) *A Dependency Parser for Variable-Word-Order Languages*. Artificial Intelligence Programs Research Report AI-1990-01, University of Georgia, Athens.

Crocker, M.W. (1996) *Mechanisms for Sentence Processing*. Centre for Cognitive Science Research Report EUCCS/RP-70, University of Edinburgh.

de Beaugrande, R., Dressler, W. (1981) *Text Linguistics*. Harlow, England: Longman.

Dyer, M.G. (1981) $RESTAURANT Revisited, or 'Lunch with BORIS'. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI-81)*, Vancouver, Canada: Volume 1.

Dyer, M.G. (1992) Scripts. Entry in *Encyclopedia of Artificial Intelligence (Second Edition)*, ed. S.C. Shapiro. John Wiley and Sons.

Garnham, A. (1983) What's Wrong With Story Grammars. *Cognition* 15, 145-154.

Garnham, A. (1987) *Mental Models as Representations of Discourse and Text*. Chichester: Ellis-Horwood.

Gazdar, G., Klein, E., Pullum, G., Sag, I. (1985) *Generalised Phrase Structure Grammar*. Blackwell: Oxford.

Gazdar, G., Mellish, C. (1989) *Natural Language Processing in PROLOG: An Introduction to Computational Linguistics*. Addison-Wesley.

Guenther, F. (1987) Linguistic Meaning in Discourse Representation Theory. *Synthese* 73, 569-598.

Hobbs, J.R. (1979) Coherence and Coreference. *Cognitive Science* 3(1), 67-90.

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

Kamp, H. (1981) A Theory of Truth and Semantic Interpretation. In *Formal Methods in the Study of Language (Volume 1),* eds. J. Groenendijk, T. Janssen, M. Stokhof. Amsterdam: University of Amsterdam.

Kehler, A.S. (1995) *Interpreting Cohesive Forms in the Context of Discourse Inference*. Ph.D. dissertation, Harvard University.

Kieras, D. (1983) A Simulation Model for the Comprehension of Technical Prose. *The Psychology of Learning and Motivation* 17, 39-80.

Kintsch, W., van Dijk, T. (1978) Toward a Model of Text Comprehension and Production. *Psychological Review* 85(5), 363-394.

Lakoff, G.P. (1972) Structural Complexity in Fairy Tales. *The Study of Man* 1, 128-150.

Lehnert, W.G., Dyer, M.G., Johnson, P.N., Yang, C.J., Harley, S. (1983) BORIS - An Experiment in In-Depth Understanding of Narratives. *Artificial Intelligence* 20, 15-62.

Mandler, J.M., Johnson, N.S. (1977) Remembrance of Things Parsed: Story Structure and Recall. *Cognitive Psychology* 9(1), 111-151.

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

Mann, W.C., Thompson, S.A. (1988) Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation. *Text* 8, 243-281.

Mikheev, A. (1996) *Domain Knowledge for Natural Language Processing*. HCRC Language Technology Group Technical Report, University of Edinburgh.

Minsky, M. (1975) A Framework for Representing Knowledge. In *The Psychology of Computer Vision,* ed. P. Winston. New York: McGraw-Hill.

Norvig, P. (1989) Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science* 13, 569-620.

Polanyi, L. (1995) *The Linguistic Structure of Discourse*. CSLI, Stanford, on-line technical report.

Rich, E., Knight, K. (1991) *Artificial Intelligence (Second Edition)*. McGraw-Hill.

Rimmon-Kenan, S. (1983) *Narrative Fiction: Contemporary Poetics*. London: Methuen.

Rumelhart, D.E. (1975) Notes on a Schema for Stories. In *Representation and Understanding*, eds. D.G. Bobrow, A. Collins. New York: Academic Press.

Rumelhart, D.E. (1980) On Evaluating Story Grammars. *Cognitive Science* 4, 313-316.

Schank, R. (1975) *SAM: A Story Understander*. Technical Report, Yale University, Computer Science Research Report #43, New Haven, Conn.

Schank, R., Abelson, R. (1977) *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Shieber, S.M., Schabes, Y., Pereira, F.C.N. (1995) Principles and Implementation of Deductive Parsing. *The Journal of Logic Programming* 24(1-2), 3-36.

Siklaki, I. (1985) Story Grammars as Models for Story Processing. In *Text Connexity, Text Coherence: Aspects, Methods, Results*, ed. E. Sözer. Hamburg: Buske.

Spencer-Smith, R. (1987) Semantics and Discourse Representation. *Mind and Language* 2(1), 1-26.

Thibadeau, R., Just, M.A., Carpenter, P.A. (1982) A Model of the Time Course and Content of Reading. *Cognitive Science* 6, 157-203.

Thorndyke, P. (1977) Cognitive Structures in Comprehension and Memory of Narrative Discourse. *Cognitive Psychology* 9(1), 77-110.

Wilensky, R. (1983) Story Grammar Versus Story Points. *The Behavioural and Brain Sciences* 6, 579-623.

Zhang Songmao (1994) Story Parsing Grammar. *Journal of Computer Science and Technology* 9(3), 215-228.