A Model of Story Generation

A thesis submitted to the University of Manchester for the degree of M.Sc. in System Design in the Faculty of Science

Mark G. Lee

1994

Department of Computer Science

Contents

Abstract

This dissertation investigates the types of knowledge and processes required for story composition and describes the implementation of a story generation prototype which illustrates these.

The failure of previous attempts in A.I. has been due to the lack of knowledge of story structure and use of discourse features in the output text.

The first problem is solved by the use of a story grammar. Despite a great deal of empirical evidence, such grammars have been argued to be psychologically implausible. The dissertation provides a defence of story grammars, arguing that critics have misconceived the implications of such grammars and misunderstood the conditions for a grammar to be psychologically real.

To solve the second problem, a mechanism is described which generates natural language text with cohesive features using a case grammar.

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Dedications

This thesis is dedicated to Betty and Arthur (Mum + Dad), and Pauline.

Acknowledgements

I wish to thank everybody involved with the System Design Msc at Manchester University.

Thanks also to my currant supervisor at Sheffield University, Dr Paul Mc Kevitt for helping me with revisions and proof reading drafts.

Thanks to Dr Mike Lebowitz for providing me with literature concerning his Universe program and also discussing various details of his program via email.

Also to Dr Scott Turner for sending me his PhD thesis.

Thanks to Zoltan Rigo, for solving various Latex technical problems.

Last but not least, many thanks to Pauline Webb for help typing the thesis and putting up with me.

Chapter 1

Introduction

The majority of work in artificial intelligence and computational linguistics has been aimed at practical, scientific problems such as expert systems and machine translation. This is to be expected. However, A.I. also has applications in the arts. Not only in the creation of works of art but also in the cognitive understanding of how aesthetic and creative processes operate.

Previous work has looked at various areas - drawing [Coh81], poetry composition [Shi72] and even Jazz improvisation [Joh91]. The area I have looked at is story composition.

I have chosen to concentrate on building a program which generates stories unaided. Previous story generators of this kind include Klein's "automatic novel writer" [Kle73], Meehan's Talespin [Mee76], Lebowitz's Universe [Leb87] and Turner's Minstrel [Tur92]. I will discuss each in detail and show that they all fall short of the task due to lack of story knowledge and structure i.e. a lack of the knowledge of what makes a good story. They also fail to use linguistic features of discourse which are required for the stories produced to be readable.

One of the first criticisms of A.I. was that a computer program cannot originate anything - it can only do what it is programmed to do. Turing [Tur63] ascribes this criticism to Charles Babbage's sponsor Lady Lovelace. To such objectors any machine which merely follows rules is only imitating creativity. However, cognitive science is based on the assumption that rules and procedures underlie human cognition - including creativity.

As Schank [Sch90] notes, creativity via rule following needn't be a contradiction - as long as we define certain behaviour as creative and can find algorithms which perform such behaviour then we can begin to develop a scientific theory of creativity. Whether such an algorithm would satisfy some philosophical definitions of creativity is questionable, however I believe that redefining creativity into the computational paradigm is an important first step. Not only might we discover new answers but just as importantly, find new questions which need to be answered.

This is not to claim my model as redefining creativity or possessing any originality, rather that it illustrates the types of knowledge and cognitive processes used in a creative process - story composition.

The central aim of the project, therefore, is to build a story generating program. The stories generated have to display the types of structure found in human authored stories and show the features of discourse found in natural language text.

The thesis has three threads - existing A.I. knowledge based approaches to story generation, the structure (or grammar) of human stories and the linguistic features of discourse. These threads are discussed in turn and finally combined to implement a model of story composition.

1.1 Existing A.I. story generators

Early story generators focused almost exclusively on the task of planning a story from a predefined setting (for example, [Kle73] [Mee76]). Typically, the initial situation and characters are either randomly selected or specified by a user and the planning system simulates the later events arising from the characters reacting to the setting.

This is insufficient - the crucial part of story generation is the choice of situation and characters. Without a good choice of both, any story produced - no matter how plausible - will be trivial and uninteresting. Knowledge of what makes a story successful is needed. This knowledge must include story structure, types of character which operate in stories and what sort of events should be included. Typically, humans write stories with a central theme or idea in mind, the actual details and step by step events come later. There is also a great deal of psychological evidence that when reading stories, at least, the structure present aids the comprehension of the text. Such structure in previous story generators has been absent.

Later work in story generation has reflected the above. For example, Lebowitz [Leb87] presents a set of heuristics for generating characters to suit specific situations and Turner [Tur92] posits the use of frame like structures to ensure that episodes within the story are structured coherently and have a point to them. However, both methods appear to underestimate the amount of structure present in story discourse.

One way of incorporating such structure is using a grammar of story structure to constrain the generation. Not only is this a practical solution due to our understanding of the theory behind grammars, but also it has interesting psychological implications.

Previous story grammars also fail to make full use of the linguistic features of discourse. Such a failure results in stories which are not only conceptually uninteresting but difficult to read and understand. Any future story generator must make use of these features.

1.2 Story grammars

The idea of a grammar of story structure isn't a new idea. Originally story grammars were used in anthropology to classify folktales from different cultures. Story grammars were adapted for cognitive psychology to explain the processing and remembering of text. The focus remained on folktales due to two reasons. Firstly, folktales exhibit story structure at its clearest. Secondly, folklore is an oral tradition and historically not set down in written form. Therefore, folktales have been adapted to be easily remembered and understood - parts of the tale which are difficult tend to be omitted in future tellings or changed. This probably accounts for the clear structure found in such tales and a strong claim can therefore be made that such structure aids comprehension and recall and therefore can tell us about how information from texts is encoded and recalled.

However, story grammars are currently unfashionable - in fact Anderson and Slator [An+Sl90] regard story grammar theory as being one of the few examples of a successful scientific theory refutation in A.I.. Possibly unfortunately for the scientific status of A.I., such claims are premature. I will consider and refute the criticisms aimed at story grammars on two levels.

Firstly, critics fail to take into account the actual claims made by the theory - for example, arguing against the idea that comprehension is purely affected by the structural factors associated with the grammar. Often critics also argue that story grammars are too function specific to be psychologically plausible - since story grammars are genre specific, the claim is that a separate grammar is needed for each type of discourse. Both objections are aimed at a straw-man. Story grammars are compatible with other comprehension mechanisms and represent a wider cognitive problem solving mechanism.

Secondly, at a higher level, critics misconceive what psychologically real means for such a grammar, mixing causal with descriptive accounts of in-the-head processing.

1.3 Discourse processing

Any user of natural language can distinguish between a set of isolated sentences and a text. The text has cohesive features such as reference and ellipsis which unify the sentences into a single entity. Such cohesion is necessary for the text to "flow" and be easily read.

Any program which generates discourse must therefore be able to use such features to avoid the output being difficult to understand. Unfortunately, this point has been hitherto virtually ignored by story generation programs which have focussed purely on the planning aspects with a detrimental effect on all the other features of good story telling.

The prototype implemented makes use of a case grammar to generate text with anaphoric features such as reference, ellipsis and substitution, and conjunction between the sentences.

1.4 Overview of the structure of the thesis

In this chapter I have briefly outlined the goal of the project - to construct a prototype storytelling program and investigate the types of knowledge required in such. In Chapter two, I will review some of the previous attempts of story generation and show that such models fail due to the points raised above. In chapter three, I will outline work done in story grammars which can be used to solve the lack of structure in such generated stories and in chapter four, I will show that such grammars are psychologically plausible as descriptive accounts of the types of information involved. Chapter five outlines the linguistic features which are missing in previous work and chapter six describes the implementation of the story generator and gives some examples of its work. Finally, in chapter seven, I will discuss the conclusions arising from this work and discuss possible future directions.

Chapter 2

Existing story generators

2.1 Klein's automatic novel writer

Rather than generate short stories, Klein's "automatic novel writer" (Klein et al [Kle73]) generates 2000 word murder novels - an excerpt of which is below -

> "The day was Monday. The pleasant weather was sunny. Lady Buxley was in a park. James ran into Lady Buxley. James talked with Lady Buxley. Lady Buxley flirted with James. James invited Lady Buxley. James liked Lady Buxley. Lady Buxley liked James. Lady Buxley was with James in a hotel. Lady Buxley was near James. James caressed Lady Buxley with passion. James was Lady Buxley's lover. Marion following them saw the affair. Marion was jealous."

[Bod77]:300

The program is simple. A cast of characters and locations are selected at random, along with random features - such as flirty, dishonest etc. The story progresses to a preset length, when only one character is randomly selected to be murdered. Every other character is said to be looking for clues and then the identity of the murderer is stated (again chosen randomly - though constrained not to like the victim). This statement is made boldly without the progressive discovery of clues etc.

It is clear that the program's stories aren't very convincing and it is worthwhile discussing why. "Automatic novel writer" has three main weaknesses - lack of story structure, lack of world knowledge and poor use of natural language. I will discuss each in turn.

The story's structure is very poor; certain events occur without consequence - for example, as the story above progressed, the affair between James and Lady Buxley was found to be completely irrelevant. This is due to the program not having any explicit knowledge about what makes a good story. A good murder story contains numerous suspicious characters, red herrings and a surprise ending which is progressively built up to by the discovery of clues. Instead, Klein's program randomly selects the characters, describes them, allows a small bit of interaction using a simple planning system, kills a character randomly and then states who the murderer was.

The actual knowledge of social relations is very poor. Later, in the story above, the butler kills James. The subplot of Marion helping the butler isn't even a possibility. Nor does the butler attempt to disguise his deed. In fact, the character has no concept of desiring to avoid discovery or providing red herrings.

Finally, the linguistic structure of the text is very obtuse - if the story is to be enjoyable, not only must it be interesting, it must also be easy to read. However, all the sentences have the same repetitive, simple structure and stand alone as statives. Real discourse, however, coheres into a single entity by the use of linguistic aspects such as anaphoric and cataphoric referents and ellipsis.

In summary, Klein's model was an early attempt to automate story composition but failed because it lacked story structure, lacked much of the real world knowledge needed and made poor use of natural language.

2.2 Meehan's Talespin

2.2.1 Introduction

Talespin [Mee76] is a program which like Klein's "automatic novel generator" generates stories. However, Talespin's stories are shorter and consist of the solution of some problem by the protagonist. The operation of Talespin is also different - instead of any topdown constraints, Talespin operates in a bottom up mode. A world is created and a protagonist is given a goal to achieve. A simple planning system is used to construct the story with each planning step and inference translated into natural language. The plan might involve other characters which help or hinder the protagonist. For example, below is a typical story created by Talespin.

> "George ant was very thirsty. George wanted to get near some water. George walked from his patch of ground across the meadow

> through the valley to a river bank. George fell into the water. George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. George couldn't get near the meadow. Wilma Bird wanted George to get near the meadow. Wilma wanted to get near George. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow. George was devoted to Wilma. George owed everything to Wilma. Wilma let go of George. George fell to the meadow. THE END."

[Mee81]:199-200

Talespin embodies the view that stories are, at the lowest level, problem solving narratives. However, as Meehan [Mee81]:203 points out, stories clearly can consist of more than this. I will first describe how Talespin works. I will then outline what is missing in Talespin's narratives.

2.2.2 Overview of Talespin

As noted above, Talespin first creates a "world" where the story takes place. This is achieved through interaction with the user. A planner is then used to achieve some goal given to a character. The solution might involve interacting with other characters who might have their own goals. Any character trait (such as whether a character is honest or sly) is also decided by the user.

2.2.3 World creation

Below is the prompt displayed to the user once Talespin is run .

> *(START)

************WELCOME TO TALE-SPIN***************

CHOOSE ANY OF THE FOLLOWING CHARACTERS FOR THE STORY

1; BEAR 2: BEE 3: BOY 4: GIRL 5: FOX
6: CROW 7: ANT 8: CANARY

[Mee81]:204

Once the characters are chosen, the sex of each gender neutral character is determined randomly. The story starts by describing where all the chosen characters live. The user must then decide how much knowledge each character has - for example, whether each character knows where the other characters are, and what props are available (if any). Finally, the user decides what problem the protagonist has to solve[1]. It is clear, therefore, that Talespin relies on a large amount of user involvement in creating the story. This involvement determines how interesting the story is going to be.

2.2.4 Problem solving procedures

Characters achieve their goals by the use of a planning system. However, story generation requires a novel perspective of planning - rather than search for the optimal solution or even a successful solution, a story generator must look for both good and bad plans. Often in stories, failure of an attempt is more interesting than success. Therefore, Talespin has very limited look ahead capabilities and characters use the first reasonable plan considered.

[1]Meehan [Mee81]:205 lists the following possible problems - thirsty, hungry, tired or horny.

To represent events and planning steps in the story, Talespin uses Schank's conceptual dependencies (CD) ([Sch72] [Sch73] etc)[2]. In addition to the eleven primitive actions used in CD theory, Meehan adds the acts of planning and wanting to deal with problem solving behaviour. There are 13 different social states which are represented as integers between 10 and -10 and represent such traits as honesty and affection. All representations are passed to a natural language generator which translates each proposition into an English past tense sentence.

In addition to the conceptual dependency primitives, there are a set of procedures which allow more complex behaviour. For example, there is a promise procedure - "promise(X,A,Y,B)" which means that Y has offered to do B if X does A. If X is dishonest, then X will agree but plan to insult Y once Y has performed B. If X likes Y, X will agree. Otherwise X will disagree (see [Mee76]:210).

Finally, Talespin uses a set of problem solving strategies which characters in the story use to achieve their goals. These are based on Schank and Abelson's D-goals [Sc+Ab77]. For example, Dcont is used if X wants to acquire Y. Talespin checks to see if X has already acquired Y. If not, Talespin gives X the subgoal of getting to Y (calling a procedure called Dprox). Once in the same location, X attempts to take Y (even if Y being held by another character) which involves the sub-procedure Do-atrans. If the attempt succeeds then Dcont is also successful.

[2] [Har85] and [Sch72] both provide concise overviews of CD theory.

Other strategies include ones for persuasion, finding knowledge and communicating with other characters.

2.2.5 Inferences

Talespin needs to make inferences to access the effect of events it generates. For each conceptual dependency primitive there are four types of inference which Talespin considers.

Type one inferences specify the previous events which could lead up to the current event. This type of inference is most commonly used in character planning. Type two inferences are the consequences of each action. These are the most common type of inference and include the reactions of other characters to the action. Type three inferences refer to the states which are preconditions for any planned procedure or action. Finally, type four inferences specify the set of events which might plausibly follow the current event or action.

Such inferences are essential for two reasons. Firstly, to allow characters to plan and respond to actions in an intelligent manner. Secondly, to ensure that the narrative is plausible and coherent - i.e. that all the events in the story proceed in a sensible way.

2.2.6 Control of Talespin

As noted above, there is initially some interaction with the user to determine what the story is to consist of. Once specified, Talespin's control cycle begins.

The problem solving procedure corresponding to the current goal of the protagonist is looked at and a plan for the current situation formulated. This plan might involve subgoals (which call further procedures) or immediate actions which are added to the narrative output. Once an action is carried out by the protagonist, the state of the world is updated. Any new inferences and details are also output.

The consequences of the action are then calculated. These are worked out by the inference mechanism (see above). The inference mechanism often generates a proliferation of facts - especially if two or more characters attempt to second guess each other[3]. Talespin copes because the knowledge used for character inference is incomplete and therefore the inferences eventually die out [Mee81]:222.

Once the protagonist performs an action, all the other characters present are allowed to perform some action to achieve their own goals (typically caused by reactions to the protagonist's behaviour). To do this, they use exactly the same mechanism as above. It is important to note that Talespin limits itself to only

[3] For example, Meehan gives an example of output from a Talespin which goes "Sue wanted to ask Peggy whether Peggy would ask Betty whether Betty would ask Peggy whether Peggy would ask Betty whether Betty would ask ...if Sue gave Peggy some honey" [Mee81]:212. Incidently the answer was "no".

single protagonist stories - while other characters can exist and interact, they only have goals in relation to the protagonists behaviour e.g. if the protagonist persuades them to do something.

The cycle repeats itself till the story finishes. A story only finishes if either the protagonist has achieved all his or her goals or cannot make any more progress.

2.2.7 Linguistic generation

The generator takes a conceptual dependency and converts it into an English sentence which is immediately output to the user. Initially, Talespin used a version of Goldman's Babel language generation program [Gol75] but this was found to be too unwieldy for the application, therefore Meehan created his own program. The principal tasks of the generator consist of putting the actor, action, object and direction of the conceptual dependency in the correct position of the sentence. Therefore, the generation is directly from the conceptual dependency to the English sentence i.e. no intermediate representation is used. As we shall see, such an intermediate representation (possibly concentrating on the syntactic aspects) can be very useful.

For each verb used by the program, there is a lexical entry which details the corresponding conceptual dependency and the conjugations of the verb for singular and plural in the past, present and future tenses[4].

[4]However, all stories in Talespin are told in the past tense.

The first step in generation involves creating the subject - verb pair of the sentence. Four parameters are used - the infinitive verb, the subject (which is included in case it needs to be embedded), the tense and a mode list which is represented as part of the conceptual dependency and therefore represents some of the meaning of the action. For example, the mode list could contain operators specifying that the proposition generated is negative or an interrogative. A conceptual dependency containing both would, therefore, be translated into a negative interrogative. Once the subject verb pair is created, further items in the conceptual dependency are placed in their correct position.

Talespin's generator is an entirely ad hoc system - it isn't based on any recognised linguistic theory. Rather, it is based on Meehan's intuitions about conjugations [Mee81]:226. Therefore, the range of sentences which are capable of being generated is limited - for example, it cannot produce a sentence of the form "It was Jim who went to New York". However, since the planning system doesn't generate anything which needs this type of sentence, this is not a problem.

The most important feature of the generator is the almost complete absence of any syntactic processing. Sentences are built up directly from the semantics of the conceptual dependencies they represent. This follows Schank's claim that syntax should have a secondary role to semantics in natural language processing (for example, see [Sc+Ri81]:11). Historically this is an interesting claim because work in natural language processing at that time dealt almost exclusively with syntax. As we shall see, both extremes have problems - any generation system which wants to make full use of the features of natural language needs to process both syntax and semantics.

2.2.8 An appraisal of Talespin

Talespin is a developed attempt at automatic story generation. The stories generated are always sensible and often amusing. As Meehan notes, stories might be more but they are certainly not less than problem solving discourses [Mee81]. However, it is clear from reading the stories of Talespin that something is missing [5]. Therefore, any future attempts at story generation must ask "What is missing from Talespin's tales"?

The most obvious problem with Talespin is that often the stories generated are uninteresting. In fact, what determines the "interestingness" of Talespin's output is the choice of situation made by the user. This is all too clear when Meehan [Mee81] talks of ensuring that a crafty fox and a vain crow are present to generate a version of Aesop's "The fox and the crow". Clearly one of the most clever things about the fable is the *actual choice of situation.* Therefore, most of the work has already been done for Talespin by the user. If left to its own devices, the stories generated can be very boring indeed, though plausibly constructed. In short, Talespin has no idea about what makes a story worth

[5] However, De Beaugrande and Colby [DeB+Co79] note the similarity between Talespin's stories and some primitive culture's folktales which deal solely with problem solving behaviour.

telling.

Secondly, and closely related to above, most stories have a clear structure - especially fairy stories of the type Talespin emulates. To begin with, a good story has a clear beginning, middle and conclusion with all the sub plots tied up. Talespin's stories have a beginning (the world description) but little else in terms of structure. The stories end when the protagonist's goal is achieved - regardless of the state of any subplots. It is clear that human authors don't do this - rather they have quite clear ideas about the storie's *structure* and *direction*[6]. There are also quite clearly defined characters and preconceived expectations of their behaviour - for example, we always expect villains to be dishonest. By comparison, Talespin, when at such a choice point, asks the user to make the decision.

Both problems are due to Talespin taking a solely world simulation view of story creation. That is, Talespin simulates a world and allows the characters and their behaviour to shape the story regardless of whether this makes interesting prose. Dehn [Deh81] argues that what is needed (and is psychologically more plausible) is a "author view" of story creation. By this view, situations are created as post hoc justifications of the events which the author wants to occur in the story. For example, if the author wants a murder to occur towards the end of the story (as in Klein's "automatic novel writer"), then the author ensures

[6] Below, I will argue and review evidence that the process of reading stories is also structure dependent.

that early in the story, situations occur which lay the premise of such an event likely. Such concerns are termed "author goals" by Dehn.

Finally, something is missing from the linguistic generation. Consider the passage below -

> "George fell into the water. George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. Wilma wanted George to get near the meadow. Wilma wanted to get near George. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow."

[Mee81]:200

While this is certainly an exciting part of the story, it is difficult to read. Though the sentences are all grammatically well formed and the events are both plausible and consistent, it is not cohesive [Ha+Ha78]. That is, the sentences don't bind themselves into a single linguistic unit - it shows none of the features which are found in texts - for example, co-reference and ellipsis. The above text reads better if simple cohesive features are added -

> "George fell into the water. He wanted to get near the valley. But he couldn't. And he wanted to get near the meadow. Wilma wanted to get near him. She grabbed him with her claw and took him from the river through the valley to the meadow."

If a story is going to be enjoyable to read then it must show these features - I detail below some psychological evidence that cohesive text is easier to comprehend and recall. This is not a trivial aspect, to quote Boden -

> "Grammar is essential to literary creativity. It prevents us from being bored by verbal repetition. Even more to the point it helps in

> subtle ways to guide us through the conceptual space presented to us by the author."

[Bod91]:173

Any reasonable model of story generation must be able, therefore, to exploit the linguistic features of discourse to create, if not atmosphere, passable natural language discourse. Without these features, even the most well thought out story will still seem artificial.

Talespin offers an important base to work in story generation. Problem solving is the minimum ingredient in narratives. However, there is also a lot more. It is clear that Talespin has no knowledge about what make a story interesting - which is the main challenge to the problem of artificial story composition. The stories created also lack structure and the linguistic features which appear in well written text. Both features have implications for how successful the story is and how easy it is to understand and recall.

2.3 Lebowitz's Universe program

2.3.1 Introduction

Lebowitz's Universe program generates stories from a different perspective than the models discussed above. It attempts to generate summaries of an ongoing soap opera melodrama.

This task has several important differences to traditional story telling. Since the melodrama is ongoing, there is no definitive end and less structure to the stories generated. Because of this, the story planner must ensure that a story line doesn't go down some dead end - for example, having characters with unattainable goals. Another important challenge is keeping the story interesting while remaining consistent and plausible.

Lebowitz agrees with Dehn's [Deh81] analysis that an author orientated approach is needed to story telling and achieves this by incorporating a planner which combines "author" and character problem solving. This is done by using a form of "opportunist" planning where the maximum number of author and character goals are satisfied in each planning step.

An example of the output from Universe is shown in figure 2.1. Decisions made by the program of the final story generated are prefixed by arrow signs. As can be seen, Universe plans stories with well motivated events.

*(tell '((churn JOSHUA FRAN)) (together JOSHUA VALERIE))))
working on goal – CHURN JOSHUA FRAN
–using plan ACCIDENT-BREAKUP P1/FRAN P2/JOSHUA THIRD-PARTY/VALERIE
working on goal – DO-DISABLE FRAN
⟶ FRAN has a spinal injury and is paralysed
⟶ FRAN doesn't want to ruin JOSHUA's life
⟶ FRAN pretends to blame JOSHUA for her malady
working on goal – DUMP-LOVER FRAN JOSHUA
–using plan BREAK-UP DUMPER/FRAN DUMPED/JOSHUA
⟶ FRAN tells JOSHUA she doesn't love him

Figure 2.1: An example of Universe's output [Leb87]:6

2.3.2 An overview of Universe

Below I present a brief overview to how Universe generates its stories. In particular, I will concentrate on how characters and situations are selected and created for use in stories and how the actual stories are planned. I will also review some of the more speculative work concerned with Universe which attempts to generate new story telling scenes by generalising old ones.

2.3.3 Motivated character generation

Lebowitz notes that the two key ingredients for successful extended story generation are consistency and coherence. A story is considered consistent if later events don't contradict earlier events - for example, a character introduced as a nice guy shouldn't later start blackmailing his neighbours. Coherence involves the idea that events should be logically explainable. That is, the reader should be able to construct a causal representation of the story. As Lebowitz [Leb87] notes, surprises are fine but unjustified surprises are very dissatisfying to readers.

To ensure well motivated behaviour by the characters, Universe specifies the characters involved before each story. Universe creates a large and arbitrary number of such characters so that most situations will have a suitable character.

To specify a character, Universe uses person-frames. These were originally developed by Schank and Lebowitz [Sc+Le79]. In figure 2.2 there is an example

Name:Liz Chandler (Liz)
Marriages: Don Craig (Don) [&MF1][1980]
Tony Dimera (Tony)[&MF3]
Relations: Husband-wife Tony Dimera (Tony)
Ex-spouses Don Craig (Don)

Stereo-types: Actor socialite knockout party-goer
General description:
Heath 8
Competence NIL
Self confidence 6
Sex Female
Age Young adult
Wealth 6
Promiscuity 3
Moodiness 4
Guile 4
Niceness 5
Phys-appearance 7
Intelligence 7
Goals (FIND-HAPPINESS BECOME-FAMOUS MEET-FAMOUS-PEOPLE)

Figure 2.2: An example Person-frame adapted from [Leb85]:486

of such a frame. The information stored in this frame is reasonably self explanatory - Liz Chandler is a rich, intelligent socialite who is married to Tony Dimurra. What is more interesting is how each person frame is created.

Universe uses a set of stereotypical person-frames which consist of default values for some of the character features. Any character created is an instantiation of one of the stereotypes with any features unspecified by the stereotype randomly assigned. In figure 2.3 there are three example stereotypes.

Some of the events generated by Universe require specific characters. To accommodate this, Universe uses a simple heuristic to generate characters to

Name	Lawer	Swinger	Waiter
type	occupation	trait	job
health			
competence	7		
self confidence	7	6	
sex			
age		Y adult	
wealth	7		4
promiscuity		7	
moodiness	3	7	
guile	7		
niceness			
phys appearance		6	
intelligence	6	7	9

Figure 2.3: Three example stereotypes (from [Leb84]:182)

order.

The stereotype closest to the specification is selected and the default values given to the character. Universe then selects any different type of stereotype which satisfies the other requirements of the specification and these default values are given to the character. This allows stereotypes to be combined - for example, if the story required an attractive and intelligent character and had the stereotypes available in figure 2.3, then Universe might create a character who was a lawyer by day and a swinger by night.

This simple method of combining stereotypes allows Universe to generate detailed yet plausible characters.

2.3.4 Plot fragments

Universe views, like Talespin (see above), story generation as a planning process. The planner tries to plan the story using basic story units, called plot fragments which provide narrative methods to achieve goals.

The plot fragments used by Universe span a wide range of levels from very abstract thematic units which unfold over the length of the story and more detailed units which describe specific events. The fragments also include goals whose sole purpose is to set the stage for later events. For example, a plot fragment might be to cause the occurrence of John double crossing Joan (perhaps to satisfy the author goal of creating bitter rivals). The expansion of this fragment would involve further subgoals which may be specific to the situation. For example, if Joan was a stock broker, John might feed Joan phony inside stock market knowledge.

In figure 2.4 there is a typical plot fragment "forced-marriage" which features an evil parent trying to force his or her daughter to remain in an unhappy marriage.

The first piece of information in the fragment (labelled "Characters") lists the roles involved in the plot fragment, followed by various constraints - for example, that the parent has a "niceness" factor less than five (out of ten). The next specification (labelled "Goals") specifies what higher level goals the plot fragment can be used to achieve - in this case the goal is "churn - keep two lovers

Plot fragment - forced-marriage
Characters - ?him, ?her ?husband ?parent
Constraints - (has-husband ?her) (the husband character)
    (has-parent ?husband) (the parent character)
    (less-than (trait-value ?parent niceness) 5)
    (female-adult ?her)
    (male-adult ?him)
Goals - (churn ?him ?her) (prevent them from being happy)
Subgoals - (do-threaten ?parent ?her 'forget-it") (threaten ?her)
    (dump-lover ?her ?him) (have ?her dump ?him)
    (worry-about ?him) (get ?him involved with somebody else)
    (eliminate ?parent) (get rid of ?parent (breaking threat))
    (do-divorce ?husband ?her) (end the unhappy marriage)
    (or (churn ?him ?her) (either keep churning or)
    (together ?her ?him) (try and get ?her and ?him back together)

Figure 2.4: A typical plot fragment used by Universe [Leb85]:487

apart". Finally, the heart of the plot fragment is specified - a partally ordered set of subgoals which must be achieved to satisfy the plot fragment. Often, plot fragments contain pieces of canned text which must be output. Each of the subgoals can potentually be satisfied by a variety of different plot fragments. This allows a wide variety of stories to be generated.

2.3.5 Planning stories

Given a set of plot fragments and specified characters, Universe uses a planning system to plot the story. It maintains a precedence graph of goals. The goals are partially ordered so that certain goals have to be achieved before others. It also uses a least commitment opportunistic planning algorithm which is shown in figure 2.5.

1, Select a goal with no unfulfilled goals that must precede it
2, find the relevant plot fragments that should achieve the goal, along with possible role bindings
3, pick one of these plot fragments
4, expand the selected plot fragment: this may include i, creating new goals, ii, modifying interpersonal relationships and/or character traits and iii, generating text from simple templates.

Figure 2.5: Universe's story planning algorithm (adapted from [Leb87]:237)

It is worth noting that Universe generates text as it plans as opposed to planning a story completely before text generation. Since the stories generated are unending, this latter approach would be impossible.

Much of Universe's power comes from the large database of plot fragments. It is often the case that several plot fragments will satisfy a particular author goal. It is therefore important that Universe makes a good choice of which plot fragment to use. Universe uses two selection criteria. The first criterion is that it selects fragments which satisfy other goals in the precedence graph (including character goals). Lebowitz reports that this method has the advantage of generating tangled and complex plot lines [Leb87]:247. The second criterion is to allocate interest values to each plot fragment so that a dramatic fragment is preferred over a dull one. If there are still multiple plot fragments then one is picked randomly.

2.3.6 Plot generalisation

Lebowitz [Leb85]:492 notes that Universe's performance depends on the number of plot fragments in its database. It is clear, therefore, that Universe in the long run must be able to acquire new plot fragments to avoid becoming repetitive.

Lebowitz suggests that one possible method is to generalise plot fragments. This, however, is quite speculative and the discussion that follows concerns mainly future work.

The main process involved in generalisation is the relaxing of constraints in plot fragments. For example, the plot fragment "forced-marriage" (see figure 2.4) has been generalised by Universe to form a new plot fragment which only requires one lover to have some spouse with a nasty parent. This allows the new fragment to be used in original situations.

While such generalisations are easy to make, it is clear that they cannot be made arbitrarily since then Universe might generalise all the constraints and render each fragment meaningless. There are two criteria for successful generalisations - the generalisation must be both believable and interesting.

Lebowitz suggests that both criteria can be maintained by the use of explanation based learning methods (for example, see Mooney and De Jong [Mo+De85]). The basic idea is to build up a causal analysis of each plot fragment using, for example, Schank's conceptual dependencies [Sch72] and then generalise while ensuring that the conceptual analysis still holds.

Inclusion of author goals must be done - for example, in "forced marriage" (see figure 2.4) the only reason for getting the jilted lover involved with another is to satisfy the author goal of complicating the situation. Any generalisation of the fragment would ideally maintain this goal.

Lebowitz [Leb85]:495 suggests that only one constraint should be relaxed at a time, thus allowing the process to be monitored and controlled.

However, at the moment such ideas for extending Universe are speculative since a method of such detailed causal analysis is still not known. In fact, the development of automatic generalisation techniques has become the focus of Lebowitz's research interests [personal communication].

2.3.7 An appraisal of Universe

Universe generates ongoing stories of the sort found in soap operas. It employs a detailed representation for both characters and events in the story world. In fact, much of its success lies in the richness of its representation. Future research involves generalising the plot fragments used in the planning system.

It is important to note that Universe avoids some of the problems of more traditional story generation programs. Firstly, because of the open-ended nature of its domain, Universe requires less knowledge about story structure - for example, it doesn't need to know about providing a climax or ending to its stories.

Secondly, Universe doesn't need to be economical about the number of characters or features it creates since due to the open-endedness of each story, most will play some part sooner or later. Clearly, this isn't the case in brief stories.

Finally, it doesn't deal with the difficult problem of natural language generation and instead uses canned text and templates to generate its output.

2.4 Turner's Minstrel program

2.4.1 Introduction

Minstrel [Tur92] tells stories about the knights of the round table. These stories are typically short. In figure 2.6 there is an example of Minstrel's output.

It is clear from figure 2.6 that the generated stories have several interesting features. Firstly, the stories have a point, unlike the meandering stories of both Talespin and Universe, each story is intended to reflect some moral or message. Secondly, the story structure generated is reasonably complex. For example, in figure 2.6 the opening paragraph serves to inform the reader that Lancelot is rash so that the events later in the story make sense.

Minstrel is also intended to be a serious psychological model of creativity. As we shall see, Turner views creativity as a process of selective recall and modification in episodic memory.

It was the spring of 1089, and Lancelot returned to Camelot from elsewhere. Lancelot was hot tempered. Once, Lancelot had lost a joust. Because he was hot tempered wanted to destroy his sword. Lancelot stuck his sword. His sword was destroyed.

One day a lady of the court named Andrea wanted to have some berries. Andrea went to the woods. Andrea had some berries because Andrea picked some berries. Lancelot's horse moved Lancelot to the woods. This unexpectedly caused him to be near Andrea. Because Lancelot was near Andrea Lancelot saw Andrea. Lancelot loved Andrea.

Some time later. Lancelot's horse moved Lancelot to the woods unintentionally, again causing him to be near Andrea. Lancelot knew that Andrea kissed a knight called Frederick because Lancelot saw that Andrea kissed with Frederick. Lancelot believed that Andrea loved Frederick. Lancelot loved Andrea. Because Lancelot loved Andrea, Lancelot wanted to be the love of Andrea. But he could not because Andrea loved Frederick. Lancelot hated Frederick. Because Lancelot was hot tempered, Lancelot wanted to kill Frederick. Lancelot went to Frederick. Lancelot fought with Frederick. Frederick was dead.

Andrea went to Frederick. Andrea told Lancelot that Andrea was siblings with Frederick. Lancelot wanted to take back that he wanted to kill Frederick. But he could not because Frederick was dead. Lancelot hated himself. Lancelot became a hermit. Frederick was buried in the woods. Andrea became a nun.

Moral: "Done in haste is done forever"

Figure 2.6: An example story by Minstrel called "Frederick and Lancelot" (from [Tur92]:133-134)

2.4.2 An overview of Minstrel

Below is an overview of Minstrel's story generation. I will concentrate on how Minstrel plans stories and uses heuristics to create original situations. I will also discuss the role of author goals in Minstrel's story planning.

2.4.3 Planning stories

Like previous work in story generation, Turner views story telling as a planning task involving solving various problems at different levels. To solve such problems, Minstrel uses case based reasoning techniques (see [Ri+Sc89]). The basic principle behind case based reasoning is that current problems can be solved by remembering the solutions to previous solutions and modifying this knowledge to deal with the current problem.

Problem solving proceeds by matching the current problem to similar past problems. The past problems are transformed to the current situation and by applying this process in reverse, a solution is found to the current problem using a previous solution. This process is elaborated below. Since there is nothing specific to story telling in this process, the same planning procedure is used to deal with author and character goals.

The apparent shortcomings of this approach to story telling should be obvious. Case based reasoners solve problems by reusing solutions to previous situations (or cases). Therefore, it seems that such a process will result in the same story being told again and again - hardly a model of creativity. To avoid this, Minstrel uses a set of adaptation heuristics which provide original solutions to problems by adapting solutions from different domains. These heuristics are discussed in the next subsection.

2.4.4 Transforming heuristics and creativity

Turner provides the following characterisation of creativity: "[t]he challenge of creativity is to find and use old knowledge in new ways to form a novel and useful solution to a problem" [Tur92]:88. Therefore, the problem is to define a process which will find and recognise useful knowledge and adapt it to the current situation in an interesting and novel way. Turner suggests that the solution is to intergrate the search and adaptation processes.

In Minstrel, the search and adaptation processes are intergrated into heuristics called "transform - recall - adapt methods" or Tram's for short. Each Tram bundles a search method with a corresponding adaptation. The transform phase takes a problem and changes it into a slightly different problem - for example, by generalising its constraints or relocating it to a different domain. The recall phase uses this transformation as an index and uses it to find similar past problems and adapts any found to the original problem. Finally, the adapt phase employs a similar transformation to the previous problem's solution to provide a solution to the current problem.

Comment : Try solving the problem in a similar problem domain.
Class : Acts.
Test : Is the problem in question an Act?
Transform: 1, Find a domain which has mappings for all the components of this problem description.
2, Translate the problem description into the new domain using the problem mapping.
Adapt : Apply the reverse mapping on the solution to translate it from the new domain to the original problem domain.

Figure 2.7: a high level description of Tram: cross-domain-solution (from [Tur92]:114)

In figure 2.7 there is an example Tram called "cross-domain-solution". There are four types of Tram - general, act, goal and state Tram's. General Tram's can be used in any situation and therefore are mainly concerned with solving general author goals. Act Tram's deal with actions by characters and so, for example, are used if Minstrel has a goal of getting a particular character to do something. Goal Tram's are concerned with scheduling goals - for example, deciding which goal should be attempted first by a particular character. Finally, state Tram's are used when the story requires some state change - for example, if a character has to change state. The Tram in figure 2.7 is an act Tram and represents the basic case based reasoning mechanism used by Minstrel.

For example, Minstrel used "cross-domain-solution" when generating the above story (see figure 2.6). The third paragraph starts with a scene where Lancelot accidently meets Andrea. However, Minstrel doesn't have any knowledge of how to cause events of knights meeting ladies and therefore used the

Tram to find a solution.

The transform phase transformed the problem “knight accidently meets lady” from the medieval stories domain to a modern stories domain, which Minstrel also knew about, and therefore formed the problem “a business man accidently meets a person”.

In the second step of the process, Minstrel used the transformed problem as an index for recall in the modern stories domain to see if this problem had been encountered before. Minstrel came up with the following story.

> “John was sitting at home one evening when his dog began scratching at the door and whining for a walk. John decided to take the dog for a walk. While they were out, John ran into his old friend Pete, whom he hadn’t seen in many years. John realised that he would never have run into Pete if his dog hadn’t wanted a walk.”

The final stage involved adapting the recalled solution (or story fragment) of using a willful animal to the original problem. This was simple to do since it only involved applying the original mapping of the current problem to the previous solution in reverse. The resulting scene was new to Minstrel and involved Lancelot’s willful horse riding to where Andrea was (see figure 2.6).

2.4.5 Author goals

Turner [Tur92]:14 identifies four aims for any (human or otherwise) author. An author wants to make any story interesting, understandable, artistic and creative. Turner argues that the final aim is achieved by Minstrel by using the above Tram heuristics. To achieve the other goals, Turner gives Minstrel three additional types of author goal: thematic, dramatic and presentation goals [Tur92]:133. Thematic goals are concerned with the selection and development of the theme or point of the story. Dramatic goals are concerned with keeping the story interesting by using author techniques such as building up suspense etc. Finally, presentation goals are concerned with how the story is presented in natural language. I will discuss each in turn.

Thematic goals

The theme of a story is the point of the story. Turner restricts Minstrel to telling stories with "planning advice themes" or Pat's for short. Planning advice themes represent concise pieces of advice about how plans should be made. The concept of themes is borrowed from Dyer's work on story understanding [Dye83].

Minstrel's thematic goals are concerned with selecting and illustrating such Pat's. For example, consider the Pat in figure 2.8. The Pat advises the reader to avoid making hasty decisions that cannot be undone. Given a Pat, Minstrel will acquire the partly ordered scenes required to illustrate each part of the Pat.

The Pat's consist of different types of information, subdivided into three categories - the type of advice, the content of the advice and the context where the advice is applicable. I will briefly describe the various pieces of information contained in these categories.

Type of advice -
Name : Hasty-impulse-regretted
Value : negative
Decision point: Plan-selection
Advice content -
Decision : ?person believes something (?belief1) that causes a goal failure for him (?goal1). This and his hasty disposition motivate him to do something irrevocable (?action1).
Connection : ?person learns something new (?state) that supersedes the evidence for his earlier belief (?belief1).
Consequence : ?person now has a different belief (?belief2) which motivates him to retract his earlier goal (?goal1) but he cannot because his earlier action (?action1) is irrevocable.
Object : ?action1
Planner : ?person
Context -
Active goals : ?goal2
Currant goal : ?goal2
Currant plan :
World facts :

Figure 2.8: A planning advice theme adapted from [Tur92]:170

The "type of advice" category defines the type of Pat represented. The name slot uniquely identifies the Pat. The value slot specifies whether the advice contained in the Pat is good or bad, i.e. whether a character following this advice will succeed or fail. The decision point slot specifies at which point of the planning process the advice applies. There are two possibilities - either plan selection or goal activation. Plan selection Pat's are concerned with advising whether or not particular plans should be chosen. For example, in the above Pat (figure 2.8) hasty irrevocable plans are considered to be bad choices. Goal activation Pat's advise on whether particular goals should be activated in certain situations.

The "advice content" category specifies the actual content of the advice. In figure 2.8, I have provided high level descriptions of each piece of information contained in this category. However, in the actual implementation, each piece is represented as a set of schemata linking the various concepts involved. The decision slot represents the actual planning choice specified by the Pat. For example, the decision slot in figure 2.8 specifies that the character should, in the event of a goal being blocked, choose some rash action. The connection slot specifies the link between the decision and the consequence slots - in this case, that the person will learn something contrary to his or her old belief. The consequence slot specifies the effects of such a decision. For example, in figure 2.8 the consequence is that the character will revise his or her beliefs and regret the hasty action. The object slot specifies the object the advice concerns which, in this case, is some hasty action and the planner slot specifies the character who follows the advice given.

The "context" category specifies information about the situation in which the advice might occur. The active goals slot represents any goals that have to be active if this advice can be applied. The current goal and plan slots represent the goals and plans (if any) associated with the particular advice. Finally, the world facts slot specifies any world fact requirements for the advice to hold.

It is clear that Pat's encode thematic information in terms of plans and goals. This has the advantage of allowing Minstrel's story planner to illustrate themes simply by getting characters to follow the advice in the Pat's and ensuring that the appropriate consequences ensue.

However, this method has two limitations. Firstly, Pat's are only able to capture themes associated with goal directed behaviour. Themes about the way the world works or human emotions aren't easily accommodated by this framework and any abstract theme that doesn't impinge on the planning process cannot be represented. For example, Turner notes that the theme "Behold the wonders of nature" [Tur92]:184 cannot be represented by Minstrel. Secondly, the representation of consequences is limited. The consequence can be seen as the justification of the contents of the value slot (i.e. good or bad advice). However, the consequence can only represent abstract stereotypical situations so that while it is easy to represent "the plan is bad because it will hurt the planner", it is very difficult to represent "the plan is bad because it is inelegent and inefficient". The latter requires some plan evaluation metric which cannot be easily represented in this framework.

Dramatic goals

The primary concern of Minstrel is to link a set of events into a story to illustrate some theme (or Pat). However, as Turner [Tur92]:221 points out, authoring involves more than stating events. Good authors use a variety of techniques to increase the drama of the stories they write. Minstrel has four such techniques - suspense, tragedy, foreshadowing and characterisation. I will briefly describe how Minstrel performs the above techniques.

Minstrel used suspense in the following excerpt of a story.

> "Later Bebe believed that he would die because he saw a dragon moving towards him and believed it would eat him. *Bebe was very scared. Bebe tried to run away but failed!"*
> (From [Tur92]:222 (my italics))

In the excerpt, Bebe is threatened by a dragon. Because Bebe is a sympathetic character, Minstrel assumes that the reader doesn't want Bebe to die. Minstrel increases the suspense by demonstrating Bebe's fear and by inventing a scene where Bebe attempts to escape but fails. This is typical of Minstrel's use of suspense.

Minstrel has two criteria for knowing when to apply suspense. Firstly, Minstrel only applies suspense when some important goal is being achieved by the character. In the current implementation of the program, the only goal important enough is self preservation so Minstrel only adds suspense to scenes where some character is in danger. Secondly, suspense is only added to scenes which are important to the story - i.e. explicitly mentioned by the Pat (see above).

Suspense is added by the planning of two scenes. First, a scene is invented to demonstrate that the character in question is afraid. Second, a scene is invented where some attempt is made to get the character out of danger but which fails at the last moment. As can be seen, both techniques were used in the above excerpt.

Tragedy is also used to engage the reader. Turner [Tur92]:227 adopts the classical Greek view of tragedy as the downfall of a character due to some innate character flaw. Therefore, Minstrel looks for three criteria when deciding whether tragedy can be added to a story. Firstly, some character must fail to achieve some goal. Secondly, the goal failure must be due to the unanticipated side effects of the character's plan. Finally, the plan must be somehow motivated by a character flaw. Character flaws are traits assigned to the characters such as pride, greed or envy.

Once a potentially tragic situation is recognised then Minstrel has one technique to create the tragedy. Minstrel invents some episode where the failed plan is caused or effects some character loved by the flawed character. For example, a loved one might die unintentionally by the hands of the tragic character. This is quite a limited technique and situations where such a technique can be applied occur quite infrequently in the stories generated. However, this has the advantage that tragedy isn't overused.

Foreshadowing is a literary technique in which story incidents are used to introduce or hint at later events in the story. By foreshadowing, the author makes the story appear to be a single whole and less contrived.

Minstrel accommodates foreshadowing as an author goal. For example, in the story in figure 2.6 the climax turns on Lancelot unexpectedly seeing Andrea and Frederick kiss. This is arranged by Lancelot's horse riding into the woods. To prevent this appearing to be contrived, Minstrel uses the same technique earlier in the story when Lancelot first meets Andrea.

Minstrel uses two criteria for deciding to use foreshadowing. Firstly, the episode involved must be an explicit part of the theme being illustrated. Secondly, the event must be original, i.e. an event which Minstrel hasn't used before in a story. This is because foreshadowing works best in episodes which aren't common place and therefore run the risk of appearing contrived.

When a foreshadowing opportunity is encountered, Minstrel looks back to earlier points of the story and tries to find an episode with the same sort of action involved - for example, involving the same set of characters and state changes. If one is found then it is modified so that it foreshadows the latter scene by using the same mechanism to fulfil its goal.

Characterisation is used to ensure that the main characters in the story are both believable and interesting. Characterisation is particularly important in stories which illustrate themes based on character traits. For a story of this sort, Minstrel must be able establish the trait to justify the story.

For example, the story in figure 2.6 revolves around Lancelot's quick temper which leads to a hasty killing. Minstrel establishes Lancelot's temper in two ways. Firstly, by simply stating "Lancelot was hot tempered" and secondly, by inventing an episode which demonstrates Lancelot's hot temper (shown in the first paragraph of 2.6).

Minstrel only creates a characterisation episode if the character and his or her trait is explicitly mentioned in the theme the story is trying to illustrate. If this is the case then Minstrel generates an author goal of generating an episode which illustrates the character trait before the main part of the story begins.

Presentation goals

Presentation goals are concerned with mapping the conceptual structures representing the story to natural language text. There are two main tasks - firstly, adding scenes to the story to ensure that the story forms a coherent whole and secondly, the actual generation of text from the underlying conceptual representation. I will cover each in turn.

Additional scenes are added to the story to ensure that the various characters and events appear to be rationally interconnected. Such scenes enable the reader to progress from one part of the story to another. There are two main presentation motivated scenes.

The first are introduction scenes. This type of scene introduces the reader to the genre, time frame and the main character of the story. The first paragraph of figure 2.6 is an introduction scene.

The second type are conclusion scenes which tie the threads of the story together. The conclusion informs the reader of the fates of the characters mentioned in the story. Unless the character in question is the main character then the outcome of the character is only mentioned if something bad happens to them. This avoids verbosity and works on the default assumption that most characters will live happily ever after.

The second task involved in presenting the story is natural language generation. Minstrel uses a phrasal generation technique which is a variation of phrasal parsing. Phrasal parsing intergrates syntactic and semantic information into a lexicon of phrases. Concepts from the story are matched with lexical entries which encode information about the surface sentence structure of the concept. For example, the structure for the concept of fighting is,

```
(act
:type fight
:actor ?actor
:to ?victim)
```

(from [Tur92]:288).

This structure specifies that a fight consists of an actor and a victim. During generation, such structures are generated with instantiated details - for example, the names of the characters involved. These are then matched with entires in the lexicon. For example, the structure above would be matched with the following entry,

```
(rap:phrase fight
(Input (act nil
:type fight
:actor ?actor
:to ?victim))
(output :subject ?author fought :object ?victim *period*))
```

(from [Tur92]:289)

The subject and object slots in the entry identify the relevant conceptual entities to use in the subject and object noun positions of the output sentence. As we shall see, this formalism bears a strong resemblance to the case grammar formalism presented in chapter 6.

2.4.6 An appraisal of Minstrel

Minstrel represents an advanced attempt at automatic story generation. It employs a form of case based reasoning to plan stories and uses a set of heuristics to modify existing stories from other domains to create original story episodes. It also uses a wide range of author goals to improve the artistic quality of the stories it generates.

The main strength of Minstrel is the large amount of detail which is added to each story - even a simple story will involve satisfying numerous author and character goals. Often such goals are satisfied by the same piece of generated story. Such complexity is possible because Minstrel satisfies most author goals in a similar fashion - by recognising the opportunity or requirement, for example, for suspense - and adding some text earlier in the story. Turner demonstrates that this simple mechanism can be used to allow a wide number of author goals and techniques.

Minstrel also employs a set of heuristics to adapt events from other stories to use in its own stories. This ability allows Minstrel a limited form of learning since each adapted event is stored and can be used in future stories.

There are, however, several problems with Minstrel. Firstly, it isn't at all clear whether all the author goals used actually contribute to the final story. For example, a single piece of text might satisfy several author goals. If the same piece of text is generated with only one author goal then the rest are redundant and add little to Minstrel's story telling ability.

Secondly, the majority of the structure of the stories generated is due to the particular theme or Pat which the story illustrates. Apart from earlier events which justify the main part of the story, each story is identical in structure to each Pat. This has the danger that the story structure generated is often simple and repetitive. The number of radically different stories that can be generated is also quite limited.

Finally, the natural language generation is based on phrasal generation. Unlike the case grammar formalism which the program I have implemented uses, Minstrel's generation module doesn't appear to be linguistically motivated. Each entry in the lexicon is specific to a particular conceptual structure and basically specifies the "canned text" which should be output. This approach has the problem that such a technique can't be used for different applications or, at least not without some modification, different genre's.

Despite these problems, Minstrel represents one of the most up to date and detailed attempts at story generation.

2.5 Summary

I have provided a review of research in automatic story generation. There are two main problems. Firstly, the stories generated either lack structure or the structure displayed is repetitive and simple. Secondly, the natural language generation of the stories is quite poor and fails to exploit the variety of discourse phenomena found in natural discourse.

In the next chapter I will describe story grammars with a view to using one as a possible solution to the first problem. In chapter 5, I will describe the linguistic features present in discourse which need to be considered to deal with the second problem.

Chapter 3

Story grammars

3.1 Introduction

Story grammars have had a mixed history since first introduced to cognitive science from anthropology [Rum75]. Initially, story grammars were hoped to be the key to finding the structure of any text. This clearly wasn't the case - the various types of discourse being too different to allow a single account of their structure. In fact, the original story grammars were intended to describe one type of discourse - simple fairy tale type stories. The use of story grammars in cognitive psychology has continued this approach for several reasons.

Firstly, because folktales demonstrate story structure at its most clear. Not only are most folk tales short and reasonably simple, they have little redundancy. If a proposition is made in the story (e.g. that the hero had a magic sword), it will always be relevant to later episodes in the tale. We can contrast this to some types of detective story which feature numerous red herrings.

Secondly, the folktale is largely an oral tradition - for example, Propp [Pro68] notes that until the 17th century, all Russian written text was of a religious nature - therefore all folktales were orally based. Folktales had to be short and to the point - unessential details in the story weren't passed on.

Such folktales also had to be well structured. As we shall see, there is substantial evidence that unstructured stories are hard to remember and that the abstract structure of a story plays a large part in facilitating comprehension and memory of the story.

One active area of cognitive psychology has been identifying how discourse information is processed and what structures are used in the storage of remembered stories.

The original work suggesting that we use schema for comprehension was Bartlett's study of memory [Bar32]. He suggested that such a concept could explain some of the reconstruction aspects of recall - for example, why certain additions and omissions are made when recalling what happened in a story. However, the description of schemata in Bartlett's analysis remained vague since the original theory didn't take into account the internal structure of stories.

Below I discuss previous work done on story grammars. I will first describe the anthropological story grammars and Rumelhart's original cognitive science model. Practically all of the story grammars used in later studies have been variations of Rumelhart's original and I will outline some of the variations. In the next chapter, I will then discuss the psychological plausibility of such grammars being used by people and some of the criticisms leveled at this approach. I believe that despite these criticisms, story grammars do in fact describe the types of information involved in discourse processing and I will provide a defence against these criticisms.

3.2 Propp's Morphology of folktales

One of the first attempts to systematically characterize stories was Propp's *Morphology of the folktale* [Pro68]. Propp intended to provide a system which could be used to classify Russian folktales into different groups. Propp made three claims - firstly, that stories are characterized by the functions performed by various characters in the story; secondly, that the number of these functions was finite and thirdly, that there is a regular pattern of functions in stories. It was this pattern which Propp used to classify tales. Propp's work contains a great amount of detail and below I will outline the various parts of this work which have the greatest relevance to my project.

3.2.1 Functions in the story

Propp identified 31 different functions which commonly appear in Russian folktales. Such functions proceed in a strict order after an initial setting which describes the situation at the start of the story. In figure 3.1 there is a summary of Propp's functions. It is clear that there is a great deal of implicit structure in the functions - for example, function 3 naturally follows function 2. Propp also notes [Pro68]:69 that an event in a story can serve several different functions.

3.2.2 Characters

Though Propp notes that such functions are the essential ingredients of stories, he also notes that such functions are allocated to seven different *dramatis personae* or character types. The seven character types are villain, donor (who tests the hero and provides magic aid), helper (who helps the hero), princess (or some other sought after person), dispatcher (who enlists the hero's help), hero and false hero (who pretends to have done the hero's deeds).

The character types reflect what sort of character performs certain functions (for example, functions 12, 13, and 14, require a donor (e.g. an old witch) and a hero (e.g, a lumberjack)). These categories are quite abstract and Propp notes that there are three possible relationships between the types and characters in the story.

Firstly, there might be a one to one matchup where each character type corresponds to a single character in the tale. Secondly, one character might perform the functions of several character types. For example, Propp notes of a tale where the woodcutter's son is dispatched to go on some adventure. Before

1, One of the members of a family absents himself from home.
2, An interdiction is addressed to the hero.
3, The interdiction is violated.
4, The villain makes an attempt at reconnaissance.
5, The villain receives information about his victim.
6, The villain attempts to deceive his victim.
7, The victim submits to deception and helps his enemy.
8a, The villain causes harm to the victim.
8b, One member of a family lacks or desires something.
9, Misfortune is made known; the hero is approached and dispatched to help.
10, The seeker agrees to or decides upon counteraction.
11, The hero leaves home.
12, The hero is tested, interrogated, attacked etc, which prepares the way for his receiving either a magical agent or helper.
13, The hero reacts to the actions of the future donor.
14, The hero acquires the use of a magical agent.
15, The hero is transferred, delivered, or led to the whereabouts of an object of search.
16, The hero and villain join in combat.
17, The hero is branded.
18, The villain is defeated.
19, The initial misfortune is liquidated.
20, The hero returns.
21, The hero is chased by the villain.
22, Rescue of the hero from pursuit.
23, The hero, unrecognized, arrives home or in another country.
24, A false hero presents unfounded claims.
25, A difficult task is proposed to the hero.
26, The task is resolved.
27, The hero is recognized.
28, The false hero or villain is exposed.
29, The hero is given a new appearance.
30, The villain is punished.
31, The hero is married and ascends to the throne.

Figure 3.1: Propp's thirty one functions in Russian folktales (adapted from [Pro68])

leaving, the father provides the son with a cudgel - the father, therefore, acts as a dispatcher and a donor. Finally, one role might be played by several different characters - this is often the case when characters die. For example, a dragon is killed in battle with the hero and therefore is incapable of pursuing the hero (see functions 21 and 22). In the dragon's place some female relative is employed, e.g. his wife.

Such character types also aid the comprehension of the story since they provide the reader with familiar characters whose behaviour can be predicted in new tales.

3.2.3 The story as a whole

In order to define a tale, Propp writes

> "Morphologically a tale ... may be termed any development proceeding from villainy [function 8a] or lack [function 8b] through intermediary functions to marriage [function 31] or to other functions employed as a denouement."

[Pro68]:93 (my insertions).

There is a general sequence to stories which roughly consists of a setting, some motivation for an adventure, several episodes and finally some conclusion. Such a sequence, Propp calls a move. Tales may consist of a single move or several constrained moves. For example, a tale might consist of two moves where the first has a positive ending and the second a negative ending. Such a pattern is seen in the common folktale featuring two daughters. The first is sent on some task and performs well and is rewarded. The second is then sent and performs badly and is punished.

Propp notes that to even this general sequence there can be exceptions. These are treated as fluctuations or inverted sequences (where the normal order of two moves is reversed). Such a view runs the risk of ignoring important patterns and sub-groups of stories. In fact, Colby notes that if the same was applied to folktales from different cultures (in his case Eskimo folktales) then "there would be many exceptions that it would be difficult to perceive anything beyond a vaguely normative regularity" [Col73]:651.

3.2.4 Summary and appraisal

Propp argues that functions are what define a story and such functions are finite and enumberable. The functions in every story are constrained by some pattern. Functions also depend on the presence of certain character types and Propp identifies seven character types found in Russian folklore.

Propp's work has remained influential in anthropology and has been developed in a number of different directions. For example, Greimas [Gre66] reformulated Propp's thirty one functions into five more abstract functions -

1, disruption of a state of equilibrium.
2, arrival and mission of hero.
3, trial of the hero.
4, task of the hero accomplished.
5, return to original state.

[Gre66]

The above functions were used by Lakoff [Lak72] to generalize Propp's work and to write the first story grammar[1].

However, Propp's work is clearly short of the explicitness needed for a computational model of story telling. The functions are loosely defined and the characters lack any formal definition (as we shall see this is a recurring problem with all story grammars and one which I will attempt to tackle). This is, of course, to be expected - Propp intended his morphology to be used as a classification system and not a generative grammar. However, before Propp's ideas can be used, they will have to be extensively reformulated[2].

3.3 Colby's grammar of Eskimo folk tales

Colby [Col73] presents a far more systematic approach to narrative characterization than Propp's account [Pro68]. There are certain similarities - for example, the component parts of the description refer to concrete events. However, Colby introduces two completely original notions.

Firstly, Colby's work provides a systematic grammar of stories. Though Propp gives a general sequence of functions in a story, each sequence is unrelated

[1]Which I won't discuss further - Lakoff's grammar was rather simple and his ideas have been further developed and clarified by later story grammars (see below).

[2]In fact, a story generator was written by Alan Dundes in 1965 [Dun65]. However, it was a simple program which joined prewritten episodic chunks of text and ensured that the names of the characters were consistent.

and exceptions are not explained and simply regarded as fluctuations, inverted sequences and transformations with no further elaboration. Colby notes that if the same was applied to his domain - North Alaskan Eskimo folktales - there would be too many exceptions and no apparent pattern. Colby found that the use of higher order abstract categories in a grammar was needed to account adequately for the variety of the domain.

Secondly, Colby introduces the concept of an eidon and distinguishes between primary and secondary eidons[3]. Primary eidons, the major events, are governed by Colby's grammar rules. However, there was no apparent pattern for secondary eidons which represent minor events and are used as plot devices to ensure continuity in the story.

3.3.1 Colby's grammar rules

For a folktale to be well formed it must contain at least one move[4]. A move consists of a sequence of categories: motivation (M), engagement (E) and resolution (R).

Rule 1 Move $\longrightarrow$ M, Resp $*$
Rule 2 Resp $\longrightarrow$ E $*$ R

[3]From the Greek word "eidos" meaning "idea" or "image"; in this context essentially an episode in the story.
[4]In the grammar rules, + is used as an And operator, | is used as an OR operator and $*$ to allow more than one of these entities, entities in brackets are optional.

Rule 2 states that a Resp (response) is a number of engagements followed by a resolution.

Each category is further split into two subcategories,

Rule 3 M $\longrightarrow$ VM | IM
Rule 4 E $\longrightarrow$ (PA) + (MA)
Rule 5 R $\longrightarrow$ (IR) + (VR)

The motivation category is split into value motivation (VM) and immediate motivation (IM). The engagement category consists of a preliminary action (PA) and main action (MA). A new engagement category is indicated when a PA follows a MA. The resolution category consists of immediate resolution and value resolution. Once a VR appears, a new IR cannot follow so there is only one R after any sequence of E categories. Any intermediate category can contain any number of eidons (see below for an explanation of the terms), but within each category there is a fixed sequence of eidons.

Rule 6 VM $\longrightarrow$ Fl | Sl | Ml

Rule 7 IM $\longrightarrow$ Vl | Bt | Sp

Rule 8 PA $\longrightarrow$ (En) + (Hs) + (Ch) + (Cn)

Rule 9 MA $\longrightarrow$ (Ak) + (Fh) + (Ru) + (Ps),
(Tr) + (Me) + (Ma) + (El),
(Sf) + (Ds) + (Dc)

Rule 10 IR $\longrightarrow$ (Vc) + (Ri) + (Po) + (Rs)
(Es) + (Re) + (Mr)

Rule 11 VR $\longrightarrow$ (Gr) + (Se) + (At)

Finally, there are three rules defining the type of move. In the Northern Alaskan sample, Colby noted that the choice of motivating eidon often determined at least one of the resolution categories. Thus, the context sensitive rules in the table below are included covering affective, effective and competitive concerns of the protagonist.

	Type	Motivating eidon	Resolution eidon
Rule 12	Affective	Bf	Mr
Rule 13	”	Sp	Po | Re | Mr
Rule 14	Effective	Wl	Po
Rule 15	Competitive	Ml	Po
Rule 16	”	Vl	Vc | Es

3.3.2 Primary eidons

Primary eidons represent the terminal entities of the grammar rules presented above; as such they are classified into three types - motivation, engagement and resolution. Each eidon has a number of variations and rather than provide an exhaustive listing, figure 3.2 shows an instance of each terminal entity grouped under the three types. Each eidon is given several specific instances - for example, Tr includes events such as roping an animal and removing a bucket from somebodys head. It is interesting to note the similarity between the eidons and Propp's functions. This suggests that certain events are universal to all stories regardless of culture.

3.3.3 Secondary eidons

Motivation eidons :
Food lacking (Fl) - the scarcity of food is highlighted.
Spouse lacking (Sl) - the need of a spouse is highlighted.
Maturity lacking (Ml) - the protagonist's youth is seen as a handicap.
Villainy (Vi) - the adversary causes harm to the protagonist or his group.
Betrayal (Bt) a member of family betrays the trust of the protagonist.
Separation (Sp) The protagonist is separated from the family.

Engagement eidons :
Encounter (En) - The protagonist encounters a being who helps him.
Hospitality (Hs) - The protagonist enters a home and is shown hospitality.
Challenge (Ch) - a challenge is given between protagonist and adversary.
Confrontation (Cn) - The evil deeds of the adversary are exposed.
Provocation (Pk) - The protagonist lures or flirts with a women.
Attack (Ak) - An animal, monster etc attacks the protagonist.
Fishing and hunting (Fh) - some normal deed is performed by the protagonist in some odd manner.
Retrieval attempt (Rv) - A person attempts to retrieve a child or possessions.
Persuasion (Ps) - The protagonist tries to persuade another to do some task.
Transaction (Tr) - The protagonist is asked a favour which is later returned.
Magical engagement (Me) - The protagonist uses magic against the adversary.
Magical aid (Ma) - The protagonist helps another with magic.
Elimination (El) - Defenceless associates of the adversary are driven away.
Struggle (St) - The protagonist fights with the adversary.
Discovery (Ds) - The protagonist discovers a sought after person.
Deception (Dc) - The protagonist tricks the adversary.

Resolution eidons -
Victory (Vc) - The protagonist wins in a situation.
Release (Rl) - Some game or treasure is released to the protagonist.
Possession (Po) - The protagonist gains possession of useful goods.
Escape (Es) - The protagonist escapes.
Restoration (Rs) - Some is restored to life.
Reunion (Re) - The protagonist is reunited with his group or family.
Murder (Mr) - The protagonist kills someone.
Group of reference (Gr) - The protagonists deeds are marvelled at.
Settlement (Se) The protagonist settles in a village.
Attainment (At) - The protagonist becomes a great chief or hunter etc.

Figure 3.2: Primery eidons found in North Eskimo folktales (from [Col73]:647-650)

Scene change :
Travel or flight (Tf) - The protagonist travels - possibly fleeing from the adversary.

Conveyance of information :
Intelligence(In) - The protagonist receives verbal information about the adversary or some danger later in the story.

Prolongation :
Obstacle or evasion (Oe) - The protagonist is delayed by an obstacle which he overcomes by magic or evasion.

Episodic transition :
Routine activities (Ra) - The narrative is interrupted by routine events before a new E - R episode or a new move.

Facilitation :
Facilitation (Fc) - The protagonist is instructed in magic or gains some skill.

Figure 3.3: Secondary eidons found in North Eskimo folktales from [Col73]:652-653

Secondary eidons appear not to be governed by any kind of rule. Their role is typically to ensure the smooth progression of events, highlight a feature of the protagonist or spice up a slow part of the tale. Secondary eidons are grouped into five functional categories - scene change, conveyance of information, prolongation, episodic transition and facilitation. Figure 3.3 shows the five types with an example of each.

3.3.4 Summary and discussion

Colby provides a far more systematic approach to story characterization than Propp. He introduces a grammar which combines context free and context sensitive rules. He also distinguishes between primary and secondary eidons.

Primary eidons are governed by the grammar and detail the main content of the story. However, rough edges in the narrative are smoothed by the introduction of secondary eidons. As we shall see this distinction is particularly useful to my story generator since it allows a combination of top down and bottom up processing to determine the story context. It is important to note that like Propp, Colby's work is very culture specific. This is at odds with later cognitive science story grammars but has the advantage of providing concrete objects as the terminal entities in the rules.

3.4 An outline of Rumelhart's story grammar

The first generalized attempt in cognitive science to describe the structure of stories was by Rumelhart [Rum75]. This was a seminal work and all later work in cognitive science has been based on this grammar. It is therefore worthwhile outlining the structure of the grammar.

The grammar consists of two sets of rules - one corresponding to the syntax and the other to the semantic structure of stories[5]. Below are the proposed rules.

Rule1 : story $\longrightarrow$ setting + episode

Rule1': allow(setting,episode)

[5] Semantic rules are marked with a '.

That is, all stories consist of a setting plus an episode. Often the setting consists of a set of stative propositions describing the domain of the narrative. The semantic rule states that the setting allows the rest of the story to proceed.

Rule2 : setting $\longrightarrow$ (state)$*$

Rule2': and(state,state,state ....)

That is, a setting is a set of co-joined propositions.

Rule3 : episode $\longrightarrow$ event + reaction

Rule3': initiate(event,reaction)

Rumelhart defines episodes as "special kinds of events which involve the reactions of animate objects to events in the world" [Rum77]:214. Episodes are seen by Rumelhart as being a central unit in stories. The use of initiate in the semantic rule is from Schank [Sch73] and is used to imply a causal relationship between an event and a reaction of a thinking being. Event is the most general category in the grammar.

Rule 4 : event $\longrightarrow$ episode | change of state |

action | event + event

Rumelhart only gives a semantic interpretation of the fourth possibility.

Rule 4': cause(event,event) | allow(event,event)

Rule 4' states that is there must be a causal relationship between events in the story.

Rule 5 : reaction $\longrightarrow$ internal response + overt response

Rule 5': motivate(internal response,overt response)

"Motivate" relates internal beliefs and emotive states to the overt behaviour of intelligent agents in the story.

Rule 6 : internal response $\longrightarrow$ emotion | desire

There is no semantic interpretation of rule 6.

Rule 7 : overt response $\longrightarrow$ (attempt)$*$

Rule 7': then(attempt, attempt,....)

Rule 7 states that an overt response consists of a number of attempts. Rule 7' states that the attempts to achieve a goal occur in a sequential temporal order.

Rule 8 : attempt $\longrightarrow$ plan + application

Rule 8': motivate(plan,application)

Therefore the plan motivates the application of itself.

Rule 9 : application $\longrightarrow$ (preaction)$*$ + (action + consequence)

Rule 9': allow(and(preaction,preaction ...),

initiate|cause|allow (action, consequence))

The application of a plan consists of three parts - a optional series of preactions followed by an action and its consequence. Rule 9' states that the temporally sequenced set of preactions allow the action which either initiates, allows or causes the consequence.

Rule 10 : preaction $\longrightarrow$ subgoal + (attempt)$*$

Rule 10': motivate(subgoal,then(attempt,attempt.....))

Rule 10 states that preactions consist of a subgoal and a optional set of attempts to achieve the subgoal which are sequentially temporal.

Rule 11 : consequence $\longrightarrow$ reaction + event

Finally, consequences need to be defined. Consequences can be any sort of event resulting from an action and consist of a reaction followed by some event.

The use of such rules allows a parse tree to be built for well formed simple narratives. Rumelhart concludes that the grammar rules are only a tentative beginning of a theory of narratives. Rumelhart also notes that the rules have difficulty in handling complex multiagent narratives.

Certain features of the grammar should be stressed. Firstly, the semantic set of rules stresses the importance of the causal links between events in the discourse. Earlier events motivate, cause, initiate and allow later events in the story. As we shall see, not only is this one of the strengths of the grammar, it is also one of the unsolved problems with story grammars in general. I will return to this point later, but it is important to note that though the grammar is intuitively appealing, it is extremely difficult to formalise concepts such as "causes", "motivates" etc, which of course are essential if the grammar is to be used in artificial intelligence.

Secondly, the terminal operators are still quite abstract from the linguistic surface form of the discourse or even its immediate semantics, and therefore empirical evidence is needed to validate that these are indeed the right sort of abstract entities found in stories.

3.5 Modifications to Rumelhart's story grammar

Most of the work on story grammars since Rumelhart's original work has remained fairly faithful and most of the modifications are merely elaborations. Below I will describe some of the typical changes and adaptations.

3.5.1 Rumelhart's modifications

Rumelhart [Rum77] further develops his 1975 grammar by adding "problem solving episodes". To quote

> "Such stories have the following structure: First, something happens to the protagonist of the story that sets up a goal for him or her to accomplish. Then the remainder of the story is a description of the protagonist's behaviour as he or she seeks to accomplish the goal."
> [Rum77]:269

Rumelhart notes that the behaviour of the protagonist resembles the sequence of actions attempted by a GPS style problem solver (Newall and Simon [Ne+Si72]).

Rumelhart supplies two schemata involving problem solving episodes - an episode schema and a toy schema. The episode schema specifies the relationship between the initiating event, the goal and the attempt to accomplish the goal. Rumelhart expresses this in terms of SOL[6]. The following is a high level description of the schema.

Episode about protagonist P.
1) Event E causes P to desire goal G.
2) P tried to get G until outcome O occurs.

The toy schema represents the internal structure of the attempt and can be described as below.

[6] SOL is a computer language designed to express semantic procedures.

Agent A tries to get goal G.
1) A selects a method M which could lead to goal G.
2) For each precondition P of M, A tries to get P until outcome O.
3) A does M which has consequence C.

Rumelhart shows that the schemata act like a grammar for stories in two ways - they allow the construction of a story tree, and they discriminate between well formed and ill formed stories. Rumelhart claims that summarizing a tree involves pruning its leaves - if we prune all the nodes below a level N, we produce a summary of detail N [Rum77]:282.

3.5.2 Mandler and Johnson

The grammar used in the experiments of Mandler and Johnson [Ma+Jo77] (see figure 3.4) is based on Rumelhart's grammar. However, Mandler and Johnson report that they found the dual structure of syntactic and semantic rules "unwieldy" and often "redundant" [Ma+Jo77]:114. Therefore in Mandler and Johnson's grammar, syntactic and semantic aspects are combined together. They also note the original grammar's failure to deal with multi-episode stories and they therefore extend this grammar to cope with multi-episode stories with a single protagonist in each episode.

In the grammar, all terminal nodes represent either a state or an event. The terminal nodes are connected by three types of relationship - AND, THEN and CAUSE. States might be external or internal - for example, the "wind blowing"

fable $\longrightarrow$ story AND moral
story $\longrightarrow$ setting AND event structure
setting $\longrightarrow$ state$*$ (AND event$*$) | event$*$
state$*$ $\longrightarrow$ state((AND state)$*$)
event$*$ $\longrightarrow$ event ((AND | THEN | CAUSE) event$*$)(AND state$*$)
event structure $\longrightarrow$ episode ((THEN episode)$*$)
episode $\longrightarrow$ beginning CAUSE development CAUSE ending
beginning $\longrightarrow$ event$*$ | episode
development $\longrightarrow$ simple reaction CAUSE action | complex reaction CAUSE goal-path
simple reaction $\longrightarrow$ internal event (CAUSE internal event)$*$
action $\longrightarrow$ event
complex reaction $\longrightarrow$ simple reaction CAUSE goal
goal $\longrightarrow$ internal state
goal path $\longrightarrow$ attempt CAUSE outcome | goal path (CAUSE goal path)$*$
attempt $\longrightarrow$ event$*$
outcome $\longrightarrow$ event $*$ | episode
ending $\longrightarrow$ event$*$ (AND emphasis)| emphasis | episode
emphasis $\longrightarrow$ state

Figure 3.4: Mandler and Johnson's 1977 grammar [Ma+Jo77]

is an external state and "Marge was sad" is an internal state. An event is any action which happens in the story. The AND operator links any terminal nodes occurring simultaneously. The THEN operator links two temporally sequential nodes and the CAUSE operator links causally related nodes.

From the grammar, it is possible to see that the simplest well formed story consists of a setting, beginning, development and ending (see figure 3.4).

Later, Johnson and Mandler [Jo+Ma80] present a further modification to the grammar. Firstly, they add base rules to characterize stories which involve multi-episode stories (with possible embedded episodes) and semantic restrictions on the use of these rules. Secondly, they add a set of transformation rules story $\longrightarrow$ setting AND episode

episode $\longrightarrow$ beginning CAUSE development CAUSE ending |
episode ((AND) episode)$*$ |
episode ((THEN) episode)$*$

beginning $\longrightarrow$ beginning event | episode

development $\longrightarrow$ complex reaction CAUSE goal path |
simple reaction CAUSE action |
development(CAUSE development)$*$

complex reaction $\longrightarrow$ simple reaction CAUSE goal
goal path $\longrightarrow$ attempt CAUSE outcome
outcome $\longrightarrow$ outcome event | episode
ending $\longrightarrow$ ending event | episode

Figure 3.5: Johnson and Mandlers 1980 grammar [Jo+Ma80])

which allow meaning preserving variations in the surface form of stories. There are two types of transformation rule - deletion rules and movement rules.

Johnson and Mandler argued that some types of nodes might be deleted from the surface structure - beginnings, complex reactions and endings. In all three cases, deletions were allowed if the removed node could be inferred from the rest of the story.

There are two types of movement rule. Firstly, "goal" movement where a goal is moved to after the attempt to achieve the goal. Such a move is marked by an explicit connection such as "because" or "since". Secondly, "beginning and ending" movement, which only occurs in multiprotagonist stories, where multi-protagonists act in concert in some episodes and separately in others. Such a

setting = state*
where state* introduces protagonist of first episode, including optional description of time and location of story.
beginning event = event*
where event causes protagonist to react.
simple reaction = event*
where event represents protagonists internal feelings.
action = event*
where event* represents nonplanful acts by protagonist caused by simple reaction.
goal = state*
where state* represents what protagonist plans to do about beginning event.
attempt = event*
where event represents planful acts protagonist carries out to achieve goal.
outcome event = event*
where event represents either success or failure of an attempt.
ending event = event*
where event represents long range consequences of development or a response by a character to development or emphatic statement of consequences.

Figure 3.6: Definitions and restrictions for the revised grammar (Johnson and Mandler 1980 [Jo+Ma80])

story typically includes simple beginnings followed by several separate developments followed by a simple ending.

3.5.3 Thorndyke's grammar

Thorndyke's grammar is represented in figure 3.7. Thorndyke's grammar [Tho77] was used to represent simple stories in psychological experiments. Though based on Rumelhart's, the grammar is purely syntactic and is independent of the story content. The highest level rule of the grammar splits stories into four parts - setting, theme, plot and resolution (see figure 3.7).

story $\longrightarrow$ setting + theme + plot + resolution
setting $\longrightarrow$ characters + location + time
theme $\longrightarrow$ event$*$ + goal
plot $\longrightarrow$ episode$*$
episode $\longrightarrow$ subgoal + attempt$*$ + outcome
attempt $\longrightarrow$ event$*$ | episode
outcome $\longrightarrow$ event$*$ | state
resolution $\longrightarrow$ event | state
subgoal | goal $\longrightarrow$ desired state
characters | location | time $\longrightarrow$ state

Figure 3.7: Thorndyke's grammar [Tho77]

The setting establishes the location and time of the narrative and introduces all the central characters of the story. The theme states the overall goal of the main protagonist. The plot is one or more episodes. Finally, the resolution is the final result of the episodes.

Thorndyke states that the grammar can be expressed in terms of frames [Min75] - "during comprehension of a story the frame for general stories produces a description of the current story by substituting real properties of the story for prototyped ones provided by the frame" [Tho77]:83.

As will be discussed below, Thorndyke experimented with story recall and comprehension and found that both correlated highly with the amount of structure (i.e. agreement with Thorndyke's grammar) present in the story.

3.5.4 Stein and Glenn's grammar

Stein and Glenn [St+Gl79] used a modified story grammar for recording chil-

story $\longrightarrow$ allow(setting,episode system)
setting $\longrightarrow$ state$*$ | action$*$
episode system $\longrightarrow$ AND | THEN | CAUSE(episode$*$)
episode $\longrightarrow$ initiate(initiating event, response)
initiating event $\longrightarrow$ natural occurrences$*$ | action$*$
|internal events$*$
response $\longrightarrow$ MOTIVATE(internal response,plan sequence)
internal response $\longrightarrow$ goal$*$|affect$*$|cognition$*$
plan sequence $\longrightarrow$ INITIATE(internal plan,plan application)
internal plan $\longrightarrow$ cognition$*$|subgoal$*$
plan application $\longrightarrow$ RESULT(attempt, resolution)
attempt $\longrightarrow$ action$*$
resolution $\longrightarrow$ INITIATE(direct,consequence, reaction)
direct consequence $\longrightarrow$ natural occurrence$*$ | action$*$ | end state$*$
reaction $\longrightarrow$ affect$*$ | cognition$*$ | action$*$

Figure 3.8: Stein and Glenn's grammar (St+Gl79)

dren's recall protocols. They found that there were three difficulties with Rumelhart's grammar for this purpose. Firstly, as Mandler and Johnson [Ma+Jo77] also did, they found the dual structure of syntactic and semantic rules difficult to use and therefore combined both types of rule in one set of grammar rules. Secondly, Rumelhart's grammar did not specify the full set of causal links that occur in each category and between episodes. Stein and Glenn modified their grammar to do both. Thirdly, they found that Rumelhart's original set of categories was too restrictive for the full range of stories they intended to use and therefore added some categories and deleted others.

As in Rumelhart's grammar, the episode is still the base unit of stories. Episodes can be connected by AND, THEN and CAUSE relations. In order to be classed as an episode, a set of sentences must contain an initial event which causes the protagonist to desire some goal, an action which can be an attempt or a consequence and a direct consequence which is either the success of failure of achieving the goal. If all three aren't present then the set is marked an incomplete episode. It is clear therefore that the structure of episodes in Stein and Glenn's account is more complex than in Rumelhart's. Stein and Glenn's grammar is shown in figure 3.8.

3.5.5 Summary of modifications

As noted above, the grammar first developed by Rumelhart has been developed in different directions. The most common change has been the removal of a dual set of syntactic and semantic rules. For some, this has been done by intergrating the two sets of rules to deal with syntax and semantics [Ma+Jo77], [St+Gl79]. Other grammars have dropped handling semantic features altogether [Tho77]. However, it is clear that semantic coherence is present in stories and therefore has to be dealt with, if not by the grammar then by some other mechanism. Such mechanisms are mentioned - for example, the schema outlined by Rumelhart [Rum80]. Though mentioned, such suggestions haven't yet been made fully explicit and as we shall see, need to be further developed before story grammars can be practically employed.

Chapter 4

The psychological validity of story grammars

4.1 Introduction

The history of story grammars in cognitive science has been mixed. According to some sources, story grammars are central to how we process any kind of discourse[1]. However, story grammars also have their critics and clearly have some very serious flaws. Below I will first outline some of the empirical evidence for story grammars and then some of the most common criticisms. Finally, I will argue that the basic premise behind story grammars is sound i.e. that some abstract knowledge, ordered in a similar way to a grammar, is used in the comprehension and generation of discourse.

[1] For such a claim see Thorndyke [Tho77]:104.

4.2 Evidence supporting the validity of story grammars

Below is an overview of empirical evidence which suggests that subjects use high level structural information, such as that contained in story grammars, to process discourse. I will first show that we employ abstract representations in processing individual sentences and ask whether the same might not be true for larger linguistic entities. I will outline what predictions story grammars entail and offer evidence backing up such predictions. Finally, I will consider whether it is reasonable to assume that discourse information is structured.

4.2.1 Abstract representations of sentence level meaning

It is clear that even at the sentence level of meaning, people remember the gist of the sentence and not the actual words used. For example, Sacks [Sac67] had subjects listen to passages containing a sentence, for example,

"He sent a letter about it to Galileo, the great Italian scientist."

After a further 0, 80, and 160 syllables of text, a test sentence was unexpectedly presented. Subjects were asked if it was the same as the previous sentence or a changed version. Three types of sentence were used - the original version, a version with the phrases switched e.g.

"He sent Galileo, the great Italian scientist, a letter about it."

and a version where the meaning was changed by swapping the rules of the characters e.g.

"Galileo, the great Italian scientist, sent him a letter about it."

When the test was performed immediately, the subjects were quite good at detecting all the changes. However, after 80 and 160 syllables, subjects had only a chance level probability of detecting the first type of change (where no semantic change occurred). Performance on sentences where the meaning was changed remained good at 80% .

This experiment shows that whatever people remember when processing sentences, the actual surface representation isn't remembered - rather a higher, more abstract representation is used.

4.2.2 What would be evidence for story grammars?

If this is the case, what evidence is there for more abstract representations of discourse and, in particular, that something similar to a story grammar is used? If story grammars are actually used then it is possible to predict four categories of evidence that should be observed [Ma+Jo77].

Firstly, well formed stories should be easier to process and recall than ill formed ones i.e. if a story consists of episodes which are causally linked in a way corresponding to a possible parse of the grammar, then it should be more easily processed than one which does not correspond.

The recall of parts of the story will be affected. For example, the elaboration of events beyond the necessary structure posited by the grammar, will be less well recalled than parts necessary to fulfill the structural meaning of the story. Also optional events in the story will be less well recalled than essential parts of the narrative.

The sequence of events in a story will be recalled in the order specified in the grammar. If a story is presented with events in any other sequence, the out of sequence events will be recalled in the grammatical order.

Finally, it is expected that stories where crucial events are missing in the original story might be recalled with the gaps filled in by added events by the subject. Furthermore, if part of the structure of the story is ambiguous, it is more likely for distortions to occur during recall. All of the above predictions have been supported by empirical evidence.

One common experiment is to ask subjects to rate how acceptable the story is. For example, Stein and Policastro [St+Po84] found that stories missing one or more constituent parts of an episode were ranked as less acceptable than stories containing all the necessary parts. More objectively, Pollard-Gott, McClosky and Todres [Po+Mc+To79] asked subjects to sort sentences from stories into coherent events. After cluster analysis, the resulting tree structures were compared to the story grammar used in producing the presented story. The correspondence between the tree structures was striking.

4.2.3 Evidence for how discourse information is recalled

Thorndyke [Tho77] used a set of experiments to determine the effect of plot structure on recall. According to the grammar, the theme of the story was always present at the start of the narrative (see figure 3.7).

In the first experiment, he presented and later asked to recall, subjects with one of four versions of a narrative. The plot structure ranged from highly stereotypical (that is the events were causally linked and well formed according to the grammar) to the minimum of structure e.g the events presented as stative propositions. All the versions had the same semantic content.

In the first version, the narrative was left intact - the theme of the story was presented at the beginning and the plot consisted of events involving the satisfaction of subgoals by the protagonist in an attempt to achieve an overall goal. In the second version, the theme of the passage was removed from the normal position at the start and inserted at the end of the narrative but the episodes in the narrative were still temporarily and causally linked. The third version of the passage was identical to the second except the theme was completely absent from the narrative. Finally, in the fourth version the content of the narrative was presented without any temporal or causal sequencing.

In addition to testing the recall of the narrative via number of propositions remembered, Thorndyke asked subjects to summarise the narrative and a recognition test was included to measure false alarm rates for distraction due to narrative type.

For scoring the recall protocols, passages were divided into propositions - defined as a clause or sentence containing an active or stative verb. During recall, paraphrases were allowed.

For comparison, Thorndyke tested subjects on a similar recall test using the same set of propositions but initially presented randomly to the subject. They were also asked to rate how comprehensible the texts were on a 1 (low) to 10 (high) scale.

The rated comprehensibility was shown to decrease monotonically as the structure of the presented narrative decreased. However, there was little difference in comprehensibility on the randomly presented propositions. There was also little difference between the fourth narrative - the description, and the randomly presented propositions.

The pattern of the recall data was identical to the comprehensibility ratings - i.e the structural relationship between propositions was a strong determinant of recall.

During the summary task, subjects were asked after the narrative presentation to summarise the gist of the story in written form. Thorndyke found that subjects were more inclined to include propositions from the narrative which corresponded with the higher levels of the story structure as indicated by the story's parse tree. That is, subjects summarised by including propositions corresponding to the central structural elements while omitting propositions represented at the lower levels of the tree.

The above experiment showed the importance of discourse structure for comprehension and recall. This result was obtained despite the semantic content of the individual propositions in differing versions of the story being identical. Furthermore, when no narrative structure was present, recall and comprehensibility were not significantly better than for the random presentation.

In the second experiment, Thorndyke attempted to investigate the individual effects of structure and semantic content. Two passages were presented in sequence but the relationship between the passages was varied. In the first condition, the semantic content was changed i.e. the propositions referred to different entities and actions but the structure of the content was kept constant. One set of entities were concrete (e.g a farmer and some animals) the other abstract (e.g populists and federalists). In the second condition, the set of entities was kept constant but the structure varied. In the third condition both content and structure varied. One interesting question is whether the repetition of structure in the first condition would prime the structural frame in the first story to improve learning of the second.

During the experiment, the subjects were asked to rate the stories on scales of 1 (low) to 10 (high) on comprehensibility, imagery (could a mental picture be formed ?) and meaningfulness (to what extent could the subject identify with the events ?). The first story was presented and then rated followed by the second story. Afterwards, subjects performed a 30 min reading task and were then asked to recall verbatim each of the stories in writing. The results showed that comprehensibility ratings were solely due to structure. Imagery ratings, however, were due solely to propositional content. No pattern of differences due to structure or semantics for meaningfulness ratings was found.

It was also found that recall improved with practice. As predicted, the recall of a similarly structured story was significantly better than the recall of a story following a differently structured story.

However, when the two stories varied in structure but referred to the same entities interference occurred, recall accuracy for the second story was significantly lower than in the other conditions.

Thorndyke explained this as the subject's difficulty in being able to draw the structural relationships of the second story since during the first story, subjects encoded and primed a structural hierarchy of the narrative. In the second story, however, subjects would have to construct a conflicting structural representation. Thus, second story recall was typically rambling and disorganised. However, subjects didn't typically confuse events from separate stories.

Thorndyke's results strongly suggest that both structure and content play important parts in both recall and comprehension. It was found that stories with a repetitive structure (such as folk tales like "the three little pigs") were easily comprehended and recalled. Furthermore, repeated structure in different stories (even stories with different content) seemed to aid recall and comprehension, suggesting some kind of priming effect.

Further evidence for the use of story grammars in recall is seen in the addition of new material in recall. For example, given a story with some structurally necessary part missing, subjects tend to fill in the gap with a structurally coherent piece (Stein and Glenn [St+Gl79], Mandler and Johnson [Ma+Jo77] etc). This suggests that subjects must be using high level knowledge of the structure of narrative to fill in the gaps of memory. This knowledge appears even to be present in children. For example, Stein and Trabasso [St+Tr82] gave children incomplete stories and asked them to generate the ending. It was found that even young children follow structural constraints of the type dictated by story grammars.

4.2.4 Evidence for how discourse is encoded

The above is strong evidence for the role of story structure in recall. However, the evidence so far says nothing of the encoding process. It is possible that we comprehend stories in a data driven or bottom up fashion - only using higher level knowledge to guide local connections between sentences. I will return to the advantages of bottom up or top down models of processing discourse below (in fact my model of story generation adopts a mixed approach). However, there is strong evidence that structure plays an important role in the encoding process also.

For example, Haberlundt et al [Ha+Be+Sa80], measured sentence to sentence reading times for multi-episode stories. The results showed that subjects took longer to record sentences at the boundaries of episodes. Haberlundt speculated that subjects "set up" a new location in memory at the start of each episode and organised the episode as some sort of macro structure at the end of the episode. Haberlundt assumed that this extra work took more time.

However, it is probable that the correct explanation is more complex. Mandler and Goodman [Ma+Go82] extended Haberlundt et al's work to examine the reading time of all major constituents of stories - i.e all the structure predicted by a story grammar. Mandler and Goodman predicted that readers use the higher level structure of the story to note the current topic so that each sentence could be interpreted according to its context.

Since stories vary in their elaboration, readers need to use this information to tell when a given unit is finished. Mandler and Goodman hypothesised that this content shift took longer to process. To test this, they measured the reading times of stories consisting of structural units containing two sentences of equal word length and, as far as possible, equal numbers of letters.

The results gave a "zig zag" timing pattern, as the subject read the first sentence slowly and the second sentence of the unit relatively faster. This validated their original hypothesis. However, it also failed to take into account other causes of slow reading[2].

For example, the introduction of new nouns is known to slow reading speed (for summary of the evidence see [Gar87]:71-75). Mandler and Goodson noted through analysis that there were no significant differences in the number of introduced noun phrases between sentences in a unit and across a unit boundary (except during the setting element - one of the few surface markers for story structure).

To find a comparison for the structural effects, Mandler and Goodman ran a control experiment ([Ma+Go82] experiment 2). The reading measurement was repeated except the data used was sentences taken out of different stories from the first set. Thus, the same sentences were used but outside of the story structure. This had the advantage of eliminating all possible syntactic, semantic or lexical influences. The "zig zag" pattern described above disappeared completely, demonstrating that the reading rate was determined mainly by the higher level structure of the story rather than local relationships between adjacent sentences or reinforced coherence.

[2] In fact there was one anomaly. In outcome embedded stories, the second sentence in the outcome was read slower than the first sentence. Mandler and Goodman offer no explanation but since this effect has been replicated, it is safe to assume it wasn't just a glich in the experiment.

4.2.5 Is discourse information structured?

Though the above set of experiments provide strong evidence for the presence of higher level units in processing, it provides no evidence that there is any ordering of units - something which would certainly be the case if a grammar was involved. Black and Wilensky [Bl+Wi79] argue that there is no evidence for such ordering - noting that subjects can certainly understand stories which are ungrammatical, either with missing units or units which are out of order.

This is clearly not convincing - as Mandler points out [Man84]:58, this is the same as claiming that because people can understand ungrammatical sentences then they cannot be using a sentence level grammar.

Perhaps a fair test would be if it could be shown that people can understand ungrammatical stories but that it takes more effort than processing well formed ones.

Mandler and Goodman provided such a test ([Ma+Go82] experiment 3) by presenting stories with out of order units to subjects and measuring their comprehension times. The stories presented were single episode stories with one sentence per unit but with two adjacent units switched. The moved units were given a temporal marker[3] so that the temporal relationships between units weren't ambiguous. Mandler and Goodman hypothesised that when the the inversion is encountered by the reader, the reader would have to infer what the missing

[3] The following are examples of temporal markers - "since", "because", "earlier".

material might be to connect the preceding sentence with the following sentence. This process would take time and therefore affect the reading rate. The subject would then read the moved unit which would also slow the reading rate if the inferred material didn't match the unit, causing the subject to backtrack.

The subjects were presented six stories, the first untampered with, the rest with a different type of unit moved[4].

In all cases, swapping the places of two adjacent units decreased the reading rate of the story. Subjects were also given an unexpected recall test. For all the tampered-with stories, the error rate was significantly higher than in the untampered-with story. Both results show evidence of the type of ordering found in grammar rules being present in stories.

4.2.6 Summary of psychological data

Above I have provided a brief overview of psychological data which has been cited to support the existence of story grammars. It is clear that regardless of the size of structure, the surface linguistic representation is not remembered - rather the gist of what was read is stored. The central assumption behind story grammars is that the gist consists of both the semantics and the syntactical structure of the discourse. It has been shown that stories which disobey story grammar rules are judged less acceptable than well formed ones [St+Po84].

[4] However, in no story was the setting tampered with.

Similarly, randomly sorted sentences are sorted by subjects in structures which resemble the parse trees created by such grammars [Po+Mc+To79].

Strong evidence for the role of structure in recall is available. Thorndyke [Tho77] has shown that well structured stories are better remembered and this corresponds with the comprehensibility rating given by the subject. During summarisation, subjects also prefer to mention events in the story which correspond to the more abstract entities in the grammar.

There is also strong evidence that the structure predicted by story grammars strongly affects the encoding of discourse knowledge. It is has been shown that subjects tend to slow their reading rate at the boundaries of categories predicted by the grammar [Ma+Go82]. Such patterns of reading rate are absent in structureless discourse. It has also been shown that unstructured discourse is more difficult to process [Ma+Go82].

In conclusion, such evidence strongly supports that the structure of discourse not only affects the processes of comprehension, recall and encoding but is also stored in addition to the semantics of the text. Such structure is predicted by and explained by the existence of a cognitive mechanism similar to a story grammar which plays a role in discourse processing.

4.3 The formal aspects of story grammars

Like any other grammar, story grammars can be classified as types of formal language. One of the most important formal aspects of a grammar is its power i.e what it can and cannot represent. Often the power of a particular story grammar is not considered by its authors. However, one direction of criticisms I will consider below against story grammars is their power [Bl+Wi79] and therefore it is worthwhile classifying the grammars so far discussed.

Chomsky [Cho65] identifies four classes of grammars - regular (type 3), context free (type 2), context sensitive (type 1) and unrestricted (type 0). The higher the type of grammar, the lower the power, e.g. a type 3 grammar can describe a smaller subset of languages than a type 2. I will briefly outline the features of each type of grammar.

Regular grammars (Type 3)

In regular grammars, all the rules must be of the form,

$A \longrightarrow B\ t$ or $A \longrightarrow t\ B$

$A \longrightarrow t$

where A and B are non terminals and t is a terminal.

Regular grammars are appealing because of their simplicity. However, they aren't very powerful. For example, a regular grammar couldn't describe a language where a symbol x is surrounded by a balanced number of brackets e.g x, (x), ((x)) ...etc.

Context free grammars (Type 2)

Context free grammars consist of rewrite rules which rewrite one symbol at a time. All rules are of the form,

$$A \longrightarrow T$$

where A is a non terminal and T is a sequence of one or more terminals. Grammars of this form are used extensively in computer languages, and context free grammars can describe large subsets of natural language. However, we probably need a more powerful formation for any natural language[5].

Context sensitive grammars (Type 1)

All context sensitive grammars satisfy the following constraint: all rules must be of the form,

$$X \longrightarrow Y$$

[5]This is still argued but is probably true - Gazdar and Mellish note that Swiss German features structures which are slightly context sensitive [Ga+Me89]:143. See also Grishman [Gri86]:20-21.

with the constraint that the length of (number of symbols in) Y is greater than the length of X. Therefore A B $\longrightarrow$ C D E is valid but A B $\longrightarrow$ C isn't. This constraint is sufficient to assure us that the grammar is recursive. Despite their power, context sensitive grammars haven't proved a useful formalism for most of the constraints found in natural language and most natural language grammars (particularly early grammars, for example Postal's grammar[Pos67]) used in computational linguistics have been context free with added features to make them equivalent in power. It is important to note that there are degrees of context sensitivity and some grammars (and languages) can be more sensitive than others.

Unrestricted grammars (Type 0)

This is the most powerful class of grammar. In this grammar, anything can be rewritten into anything. This is, of course, a mixed blessing since as cognitive models, unrestricted grammars may be implausibly powerful.

The question remains - where do the story grammars discussed so far belong? As noted above, all of the rules in Stein and Glenn's grammar [St+Gl79] (or see figure 3.8) are composed of either a terminal node or a terminal node and one or more non terminals. Therefore, it doesn't allow any self embedded propositions, which is both necessary and sufficient for the grammar to be a regular grammar (Type 3).

All the other story grammars have a self embedding property but with the exception of Johnson and Mandler [Jo+Ma80] are context free (Type 2). Due to the additional constraints added to the grammar by Johnson and Mandler (see figure 3.6), the grammar is an unrestricted grammar (Type 0). As we shall see, the power of this grammar can be used to express doubts about its psychological plausibility.

4.4 Criticisms of story grammars

Despite the popularity of story grammars in the late 70's and early 80's, there remained critics. Two critiques need particular mention - Black and Wilensky's [Bl+Wi79] and Garnham's [Gar83]. I will outline both below. Black and Wilensky's critique I believe can be resisted, and I will provide such a defence. However, Garnham points to deeper difficulties which I believe require revisions to the original theory.

4.4.1 Black and Wilensky's critique

Black and Wilensky [Bl+Wi79] criticise the use of story grammars on three counts. Firstly, they claim that the formal properties of the grammars are inadequate. Secondly, that the coverage of the grammars includes ill formed stories and excludes some well formed ones. Thirdly, they claim that story grammars can play no role in a psychologically real account of comprehending discourse. I will outline the three objections in order.

The formal properties of story grammars

As noted above, any grammar can be classified as regular, context free, context sensitive or unrestricted according to its power. Stein and Glenn's grammar is classed as regular (the weakest class), Rumelhart's [Rum75], Thorndyke's [Tho77] and Mandler and Johnson's [Ma+Jo77] grammars are all context free and Johnson and Mandler's [Jo+Ma80] grammar is unrestricted.

However, what does this classification mean in terms of ability to classify aspects of stories? Consider the possibility of embedded episodes. According to Black and Wilensky, stories with self embedding stories are quite common - in fact Rumelhart [Rum75], Thorndyke [Tho77] and Mandler and Johnson [Ma+Jo77] all use the same self embedding story as an example - called the old farmer story. In this story the old farmer is trying to get his donkey into the shed (overall goal) by getting his dog to bark at the donkey (1st subgoal embedded in overall goal), to do this he must get a cat to scratch the dog to bark at the donkey (2nd subgoal embedded in the first) and so on. Such stories show that a story grammar must have a self embedding rule and since any regular grammar cannot in principle have such a rule, any story grammar must be of a more powerful type. This, of course, shows that Stein and Glenn's grammar isn't sufficient.

If regular grammars aren't adequate, how about context free ? A classic example of the limitations of context free grammars is their inability to represent discontinuous constituents i.e when a constituent is interrupted by another constituent but continues later.

In a story domain, this would mean an episode begins, then is interrupted by a second episode and then finishes. Black and Wilensky [Bl+Wi79]:218 provide an artificial example but argue that such stories are present in folk tales - they cite "The Arabian nights" [6], "The Decameron"[7] and "The Canterbury tales"[8]. Thus, argue Black and Wilensky, context free grammars are also not adequate.

Are context sensitive grammars adequate then? Not so, state Black and Wilensky. As noted above, all context sensitive rules can only rewrite one symbol at a time and the left side must always be less or equal to the right side. Black and Wilensky note that deletion and reordering rules are necessary since even simple stories often omit details (such as internal responses) and reorder events. However, due to the above constraints, deletion rules cannot exist due to the second constraint and reordering rules cannot exist due to the first constraint.

It appears, therefore, that story grammars must be unrestricted in power and the only candidate offered is Johnson and Mandler's transformational gram-

[6] In the "Arabian Nights" Scheherazade avoids her murder by telling interesting stories which are unrelated to her plight.

[7] In "The Decameron" aristocrats escape the plague by hiding in a castle - they amuse themselves by telling stories.

[8] In "The Canterbury Tales" pilgrims tell stories to make their journey short.

mar [Ma+Jo80]. However, Black and Wilensky cite Wexler and Hamburger (in Culicover [Cul76]) as proving that transformational grammars are unlearnable except under a set of artificial conditions. Therefore, the future for story grammars appears bleak indeed (but see below).

The coverage of story grammars

Ideally, any grammar should have two features - firstly, it should reject all ill formed entities and secondly, it should accept all well formed entities. That is, any grammar should cover the set of all correct entities and nothing more. Black and Wilensky argue that all story grammars fail on both counts.

In the first case, Black and Wilensky give the example of a procedural exposition i.e a text that instructs the reader how to do a task - not a story but rather a set of actions to achieve some goal.[9] Black and Wilensky give a procedural exposition [Bl+Wi79]:222 on how to catch a fish, which contains a setting, theme, plot and resolution and thus would be accepted by Thorndyke's grammar.

In the second case, Black and Wilensky look at Johnson and Mandler's transformation rules. The rules allow deletions of three kinds - beginnings, complex reactions and endings. The reaction deletion rule claims that complex reactions may be deleted from any episode if the immediately prior beginning is present and as long as the deleted reaction can be inferred from the context. Black

[9] Graesser [Gra78] proposes a grammar for this kind of discourse in the same spirit as a story grammar.

and Wilensky however, counter that all that is needed is if the entity can be inferred - the syntactic constraint is irrelevant. Johnson and Mandler claim that attempts cannot be deleted. However, Black and Wilensky give the following example,

> "John needed a book from the library and it was soon in his possession."
>
> [Bl+Wi79]:222

Therefore, the rules governing deletion are purely semantic and depend on the reader's knowledge rather than any structure in the story. Assuming the above example is valid and Johnson and Mandler's grammar doesn't accept it, then the only conclusion is that the grammar falls short in its coverage.

The explanatory power of story grammars.

Finally, Black and Wilensky take issue with the explanatory power of story grammars. As a model of comprehension, they argue, the story grammar proposes a circularity. The assumption is that a story grammar would aid in processing a story by matching the propositions of the story against the rules of the grammar. Then, the associated semantic rules (as in Rumelhart's grammar [Rum75]) would be used to build the interpretation of the story.

However, as Black and Wilensky [Bl+Wi79]:227 point out, matching the propositions to the rule entails already knowing what category each proposition represents, which is what the story grammar is meant to provide in the first place.

In conclusion, Black and Wilensky argue that any viable story grammar must be a type 0 unrestricted grammar - since only Johnson and Mandler's grammar is the only grammar with this power then it is the only viable one. However, such a grammar is psychologically unlearnable and cannot therefore be part of a psychological account. Secondly, no grammar captures the full subset of stories; this is due essentially to certain features of stories being purely semantic and not captured by grammar rules. Finally, the model posited by story grammars in comprehension involves comprehension before using the grammar - in short, a circularity.

Replies to Black and Wilensky

Some commentaries have stated that Black and Wilensky's critique buried story grammars as a research programme. This is definitely not the case. In fact, the critique prompted a set of forceful rebuttals. Rather than outline each in turn, I will provide a composite defence of story grammars drawn from the rebuttals, dealing with each of Black and Wilensky's objections in turn.

The formal adequacies of story grammars

Black and Wilensky object that most of the story grammars presented are formally inadequate due to the omission of certain rules, for example, Stein and Glenn's grammar cannot deal with embedded episodes. However, as Black and Wilensky themselves note, [Bl+Wi79]:219, it is a simple case for such grammars to add additional rules to cope with such aspects[10] and increase the power of the grammar.

However, this runs the risk, according to Black and Wilensky, of being psychologically unlearnable - unless the grammar is innate.[11] Mandler and Johnson [Ma+Jo80], in particular, argue against this, insisting that Black and Wilensky have misunderstood Wexler and Hamburger's work and its report by Culicover [Cul76]. Rather these authors claim that "certain types of transformational grammar could not be learnt by a child with a *particular set* of processing constraints, given *certain* limited kinds of input (any without semantic information) and a *particular criterion* for learning" [Ma+Jo80]:309 (their italics). In fact, Culicover himself states "it is quite clear that transformational grammars are learned by children" [Cul76]:291.

However, it is possible that the formal aspects of story grammars are irrelevant to their claims as psychological models. Rumelhart points out that what

[10] Johnson and Mandler [Jo+Ma80] did in fact add additional rules.

[11] Clearly, such knowledge isn't, since cultural differences are apparent - for example, Kintsh and Greene [Ki+Gr78] found that American children can remember Grimm fairy tales easier than Apache Indian stories.

makes a grammar psychologically interesting is that it "proposes an analysis of the constituent structure of a linguistic unit" [Rum80]:314. By a similar token, there probably won't ever be a grammar of English which will generate all and only valid sentences. However, what matters is that the grammar can generate an interesting subset. Story grammars are interesting because they identify which elements go together to form higher units and what these higher units might be.

Finally, Mandler and Johnson disagree with Black and Wilensky's example of stories containing stories embedded within - a feature not accounted for by any story grammar. Mandler and Johnson argue that examples such as "The Arabian Nights" are collections of stories placed in an amusing framework rather than a single story. The stories themselves can be successfully analysed by existing story grammars.

The coverage of story grammars

Black and Wilensky's first claim is that story grammars will accept ill formed stories and non stories - for example, procedural descriptions. However, Black and Wilensky adapted a procedural convention from Graesser [Gra78] by adding the traditional components such as setting and ending etc. That a discourse, adjusted to resemble a story should be treated as a story is to be expected. For Black and Wilensky's argument to work, they would have had to find an example discourse which, though not a story, would be accepted by a grammar without adjustment - something which they haven't done.

Secondly, Black and Wilensky attempt to show that Mandler and Johnson's transformation rules disallow some kinds of valid deletion and movement due to the rules being based on syntactic rather than semantic features.

To show this, Black and Wilensky use a modern example (see above). However, Mandler and Johnson object to this. The story grammar is intended to model traditional folklore. As Mandler and Johnson point out, for the example to be valid, Black and Wilensky would have had to use a traditional folktale. While purely semantic knowledge may be used in real situations, the domain in question, is an oral tradition - the point of studying such a domain is that for a story to survive by oral transmission it must have a clear and easily recognizable structure[12]. Therefore, it is quite likely that such discourse has syntactic deletions and movements. If such a story could only be interpreted according to the subjective world knowledge of the reader; then it is probable that it wouldn't be transferred unchanged through generations.

The explanatory power of story grammar

Black and Wilensky point to a circularity in the model presumably posited by story grammars. However, most of the workers in story grammars would dismiss such a model as a strawman. Black and Wilensky make the mistake of

[12] See above for further discussion.

assuming that only top down analysis is possible. However, if we allow bottom up methods to aid the comprehension, the model appears much more plausible. For example, Rumelhart [Rum77], [Ru+Or77] suggests that schemata can be used to capture the context and category of propositions in the text while the grammar selects the possible schemata to be used and ensures that the text is well formed.

To summarise, Black and Wilensky's objections carry little weight. However, this paper is important in that it helps focus on how such a model of comprehension could be made explicit. Moreover, as we shall see, some of the objections raised, suggest more powerful objections.

4.4.2 Garnham's critique

Garnham [Gar83] regards Black and Wilensky's criticisms as misguided. Rather than worry about the formal power of such grammars, the important question, he argues, is how such a story grammar could be used at all. Garnham notes that in the existing literature, there is little distinction between the grammar and the parsing process involved. Moreover, Garnham questions whether the analogy between sentence grammars and story grammars is a useful one. In its place he outlines an alternative model of discourse comprehension based on referential continuity and plausibility.

Garnham notes that despite the various strategies involved in sentence parsers, such as top down versus bottom up processing, all sentence parsers rely on certain universal features of natural language.

In particular, Garnham argues that every lexicon has a finite set of lexical items and a finite and small set of lexical categories - for example, noun, verb, adj etc. Therefore, the parser can assign each word a lexical category by lexical lookup (eliminating ambiguities using the grammar). Without the finite nature of the lexicon in its number of entries and categories, sentence parsers could not work in this manner.

It is clear that story grammars are intended to work in the same way - that is, natural units, in this case propositions or episodes, are to be given categories which can be used by the parser to build the higher structure of the narrative. However, as Garnham points out, the set of propositions or episodes is not finite. Rather, the set must be defined recursively since propositions can be embedded. For example, the proposition,

Henry knows wilma owns some jam

can be represented as

knows(Henry,owns(wilma,jam))

where the statement that wilma owns some jam is itself a proposition. Further embeddings can still be understood. For example,

Sherlock believes that Mary knows that Bill is guilty can be represented as,

believes(Sherlock,knows(Mary,guilty(Bill)))

Such sentences might, however, be harder to process. It is clear that such propositions can be endlessly embedded and therefore there can be no equivalent lexical lookup for propositions. This is not to claim, as Black and Wilensky do, that the category of a proposition could not in principle found without reference to other sentences in the narrative but that it could not be found by lexical lookup as in the case of sentence parsers.

The problem is that each syntactic category depends on its semantic content. For example, only "Henry was happy" and not "the ball burst" could count as an internal response, due to its semantic content alone. Not only has no plausible model been suggested but the problem itself has not even been discussed by the story grammarians.

4.4.3 Garnham's model of discourse comprehension

Garnham and Johnson-Laird [Gar83], [Ga+Oa+Jo82], [Gar87], [Joh83] have constructed a model of discourse comprehension which they argue avoids the problems discussed above, based on Johnson-Laird's "mental models". Essentially, a necessary and sufficient condition for a set of sentences to be classed as a discourse is if it is possible to construct a single mental model from it. It should be noted that the theory tries to account for all types of discourse.

Garnham et al's theory is a unified theory within the mental models framework.

I will first outline the theory and some of the empirical evidence used to support it. I will then question if it constitutes a valid alternative to the framework proposed by story grammars.

Mental Models

A mental model is a model of reality. According to Johnson-Laird [Joh83], humans understand by actively building such models [13]. Such models rely on three principles. Firstly, each entity is represented by an individual token in a mental model. Secondly, the properties of entities are represented by the properties of their tokens. Finally, any relation between entities is represented by an analogous relationship between the corresponding tokens. For example, the sentence -

> "The circle is on the right of the triangle"

could be represented in a mental model by the image of a triangle next to a square.

One set of entities might be ambiguous - giving rise to several possible models. For example, consider the following set of sentences -

> There are three things.
> Two are white.

[13] Johnson-Laird discusses the use of such models to explain psychological aspects such as syllogistic reasoning, language, mental imagery etc.

One is black.
Two are hexagonal.
One is triangular.

The above set allows more than one model to be built. For example, one representation could be

Thing (white, hexagonal).
Thing (white, hexagonal).
Thing (black, triangular).

Stenning [Ste78] notes that such a set is psychologically difficult to process due to this indeterminacy rather than any linguistic difficulties, since more than one mental model has to be entertained.

Mental models can be used in reasoning as follows - first, a single mental model is constructed of the premises available. This model specifies all explicit relations known to the reasoner. Second, a putative conclusion is formed, i.e. some relation which, though not specified in the premises, is entailed by the model constructed. Third, this new premise is tested by ensuring that no other model can be validly constructed which contradicts this conclusion.

In simple reasoning tasks, it appears to be relatively easy to explore alternative mental models to ensure the validity of a conclusion. However, with complex reasoning such a search is more problematic and therefore errors are possible[14].

[14] Such models therefore make claims about what kinds of error occur. E.g. Johnson-Laird [Joh83]:97-110 explains certain types of failure at syllogistic reasoning by this means.

Mental models of discourse

Mental models of discourse, therefore, are built up, sentence by sentence, with each further sentence constraining and embellishing the actual model. The discourse is said to be well formed, if at the end of the discourse, only one candidate mental model is available. Therefore, ambiguities such as anaphoric phrases cause difficulties which must be resolved.

There are, according to Garnham et al [Ga+Oa+Jo82], two principal factors which influence comprehension - referential continuity and plausibility. A text is referentially continuous if all its sentences refer to a continuous set of objects and its anaphoric phrases can be resolved - possibly using inferential processes such as bridging [Cla75] but preferably with the references resolved as the passage unfolds. For example, below, the first discourse is referentially continuous while the second isn't since the second proposition introduces two new entities - “Stephen” and “Alice”.

1, John loved Joan.
But Joan loved Stephen.
Stephen was married to Alice.
2, John loved Joan.
Stephen was married to Alice.
Joan loved Stephen.

The second factor is plausibility. Discourses which describe sensible events are easier to process than strange or nonsensical events - even if referentially continuous. This condition is clearly present to some degree in story grammars which constrain certain types of event to follow one another. However, Garnham et al view plausibility from quite a different perspective. In their explanation, sensible stories are more understandable because people use their own folk psychological knowledge of goals and motives to understand them. Therefore, "goal structures can be used as *"comprehension and memory aids""* [Gar83]:151 (my italics).

Garnham [Gar83] argues that such a model doesn't need to use any kind of lexical look up, which Garnham assumes a story grammar would need.

> "Once the parallel between sentence processing and story processing is given up, there is no need to assume that the role of a proposition in a story need be defined. Furthermore, the identification of its role can be achieved by inferences working from either stored knowledge or the proposition itself."

[Gar83]:151-152

However, Garnham is no more specific about the actual proposition identifying mechanism but suggests that past A.I. work in modeling plans (Fikes and Nilsson [Fi+Ni71]) and plan recognition in natural language (Bruce and Newman [Br+Ne78]) could be useful.

Garnham allows that we have some knowledge particular to folktales but insists that it is mainly nonstructured - for example, that animals can speak in such tales.

Furthermore, Johnson-Laird (cited in [Gar83]), claims that the only possible evidence for story grammars would be if it were shown that folk tales were encoded and recalled in different ways to other forms of discourse[15].

4.4.4 Psychological evidence for Garnham et al's model

Rather than outline all the possible evidence for the use of mental models in comprehension[16], I will just mention evidence supporting the concepts of referential continuity, plausibility and determinancy. However, it should be noted that much of the evidence so far cited in support for story grammars can be used (and has been by Garnham et al) to support the use of mental models. As we shall see, however, the concepts above have a direct relevance to the actual discourse generation process implemented in my program.

Referential continuity

In actual fact, the first mention of referential continuity was made within the framework of story grammars by Kintsh and Van Dijk [Ki+Va78].

Ehrich and Johnson-Laird [Eh+Jo82] investigated the effect of referential continuity on comprehension of discourse specifying the spatial positions of objects. In the experiment, they provided either a referentially continuous or discontinuous text and measured reading time and, after presentation, performed a recall test which involved asking the subjects to draw a spatial map of the objects.

[15] As we shall see, this is too strong a test and is due to an ambiguity of what it means to be psychologically real.

[16] See [Gar87] for an extensive overview.

For example, below are two discourses used in the experiment.

Referentially continuous -
The knife is in front of the pot.
The pot is to the left of the glass.
The glass is behind the dish.

Referentially discontinuous -
The knife is in front of the pot.
The glass is behind the dish.
The pot is to the left of the glass[17]

Ehrich and Johnson-Laird found that reading time was significantly higher for discontinuous texts. They also found that there were significantly more errors during recall.

Plausibility

Plausibility was, unsurprisingly, found to affect comprehension. Freeman and Johnson-Laird performed an experiment [Joh83]:375-377 where subjects were given passages and then unexpected recall tests. The passages were originally normal stories, however all the verbs were first removed. In the zero order condition, verbs were replaced by a separate panel of subjects who were only allowed access to the sentence to be filled in. In the third order condition, the subjects were allowed to see the preceding two sentences before filling in the verb in the third sentence - thus ensuring a corresponding increase in plausibility. Adjectives were added to each of the stories to ensure similar passage length.

[17] These discourses are from [Gar87]:168.

Freeman and Johnson-Laird found a striking correspondence between recall accuracy and plausibility. Recall as expressed as a ratio of length of recall protocols to length of text use was 0.75 for original stories, 0.68 for third order stories and 0.58 for zero order stories. There was a similar trend in rated comprehensibility.

Determinancy

Mani and Johnson-Laird [Ma+Jo82] tried to ascertain if determinancy[18] affects comprehension. They presented two types of discourse. The first type were determinate i.e they were compatible with only one spatial layout and therefore one mental model. However, the other type allowed several interpretations. Consider the following example.

> The bookshelf is to the right of the chair.
> The chair is in front of the table.
> The bed is behind the chair.

The bed could be either in front or behind the table. Mani and Johnson-Laird found that during unexpected recall, the meaning of the determinate descriptions were better remembered than indeterminate ones, but in actual verbatim recall of the passages, the indeterminate passages were better remembered. This is consistent with the explanation that indeterminate descriptions are held in memory in their linguistic form while determinate ones are held as a mental

[18] Determinancy holds if only one mental model can be constructed from the discourse.

model.

4.5 A reappraisal of story grammars

4.5.1 Introduction

Garnham's critique is powerful because not only does it suggest some very serious objections to story grammars, it also outlines an alternative model for discourse comprehension.

However, the critique is also mistaken. Firstly, the model presented is perfectly compatible with a story grammar component and rather than competing with the story grammar framework, could plausibly complement it in comprehension. Secondly, Garnham et al underestimate the role structure plays in discourse comprehension. Thirdly, Garnham's objections to the analogy between sentence processing and discourse processing restricts unfairly the possible mechanisms used in category recognition. Finally, there is some confusion about what constitutes a psychologically real grammar. I will elaborate on these points below.

Taken together I believe they show that story grammars still have a place within cognitive science, however it is important to note that Garnham's critique highlights some problems which though not in principle impossible, have not been dealt with sufficiently for a computational model to be built.

4.5.2 The compatibility of Garnham's model with story grammars

In Garnham's model, a single mental model is built up of the discourse, in a bottom up fashion with word meanings recognized first. The word semantics are combined to form sentence meanings which are used to constrain the mental model. A discourse is well formed if, after reading, only one mental model can exist. There is no top down expectation based processing[19] but the existence of referential continuity and plausibility are used to process the text in a piecemeal fashion. Both notions are concerned with the content of new sentences and their reference back to previous sentences. Therefore, no structure is used or expected by the model. This, of course, means that both notions are not concerned with the same features of discourse as a story grammar model. Since no story grammar theory suggests that structure is all there is to discourse processing, it appears that both models are entirely compatible.

Therefore, the only claim a story grammar theory need justify is that in addition to other mechanisms, some structure processing component is present in discourse comprehension and such a component is *structured* in the same way as a story grammar. Above I presented empirical evidence for the existence of such a component. However, Garnham maintains that such evidence can be adequately explained by a hierarchical structure of goals or plans [Gar83]:181[20].

[19] Such top down expectation might be found in a mechanism using a story grammar.
[20] See also Black and Bower [Bl+Bo80] for this view.

Thus, any story can be thought of as complete as long as it is referentially continuous, plausible and all the protagonist's goals and plans are fulfilled. While this is plausible for some kinds of discourse, it is not true in well formed stories. Consider the story below -

> Henry wanted some food.
> Henry thought he had some food in his larder.
> However when he checked, there was none.
> Unknown to Harry, Larry the wolf had crept in and ate the food.
> Harry went to the market and bought some food.
> Harry was happy.
> The end.

In the passage, all the goals and plans of both Harry and Larry are fulfilled and the story is both plausible and referentially continuous. However, it clearly isn't a good story - the villain of the story gets away. During reading such a story, there is a strong feeling that by the end of the tale, each character will get his or her just deserts - this is regardless of the actual plans and goals of the characters. For example, it is quite common in some folktales for good deeds to be rewarded and bad deeds punished purely by chance[21].

The above example also shows that any model of discourse processing which avoids high level structural knowledge cannot account for story generation. It is implausible that humans compose stories solely by the criteria of referential continuity and plausibility. If this were the case, then the stories produced

[21] A famous example is "Little red riding hood" where the wolf is punished only after the ad hoc arrival of a lumberjack who plays no other part in the tale. It is possible that this happy resolution was tagged on to the original tale, to conform to the structural regularities of similar tales.

would be more often than not, pointless and uninteresting in exactly the same way as the stories produced by Tailspin are (see chapter 2).

Clearly, writers have a great deal of knowledge about what constitutes a good story. The need for such structure in generation has been widely noted. For example, Schank [Sch90], argues that story creation involves selecting an abstract story skeleton - for example, "boy meets girl", "boy loses girl", "boy gets girl". Once selected, parts of the skeleton are filled in with more concrete details.

Garnham's main objection to story grammars is the analogy between sentence and story grammars. As noted above, he notes that a sentence grammar relies on a finite lexicon of words - something which is implausible for the propositions found in discourse.

This argument makes the mistake of assuming that a story grammar would need to operate using a lexicon lookup mechanism. Clearly, any grammar must have a limited number of categories of entities which can be allowed as terminals in the tree. As we have seen, in story grammars there are typically episodes, reactions, settings and resolutions. However, there is no need to assume that there is a need to use lexical correspondence. For example, following the suggestion of Thorndyke [Tho77], suppose that each terminal node of the grammar represented a frame. Possibly we can have a set of frames for each abstract category. For example, frames denoting punishment or reward for good deeds could be grouped under the resolution category.

Comprehension could proceed as follows - each sentence could be interpreted and a mental model built. However, after each addition to the model, a check could be made to see if the current model could be matched to one frame in an expected category. Trying to match the current mental model to all the frames in the knowledge base would, of course, be prohibitively costly, and therefore the grammar could be used to select which categories should be tried first.

Discourse could be processed as follows. Each sentence is interpreted and used to build up a representation of the current episode (possibly as a mental model). This representation is further constrained till it can be matched by one of the frames held in the knowledge base. A story grammar could be used to guide which frames are used to match up with the representation.

This explanation is attractive because it explains why poorly structured stories are difficult to process. The poor structure either disallows any frame match up or causes the grammar to chose the wrong category of frame for the current representation.

As noted above, Garnham and Johnson-Laird argue that the only sufficient evidence for the existence of a mental story grammar would be if certain types of discourse (e.g. folktales) were remembered and recalled in a different way to other types of discourse. This implicitly assumes that the type of knowledge embodied in story grammars could only be used in folklore comprehension. For example, Garnham writes "no experimental evidence favouring a special story processing mode has yet been reported" [Gar83]:151. Clearly such a mode is biologically implausible, but Garnham underestimates the possible role structural information of the type encoded in story grammars can play in other cognitive domains.

For example, Gick and Holyoak [Gi+Ho83] were interested in how knowledge from one domain could be used for problems in another domain. One famous example of this is Duncker's radiation problem [Dun45]. In this, the goal is to destroy a cancer. Powerful X rays are available, but the power needed to destroy the cancer would also destroy healthy tissue. The solution is to direct several weaker X rays at the cancer from different directions. This simple solution is often difficult for subjects to find and Duncker tried presenting a similar problem to aid the subject's reasoning skills - the military problem. In this problem, the aim is to storm a well defended castle. A large army is available but any march by such an army would be spotted and would alert the castle's guards. The solution is to split the army into several units which can approach the castle in different directions.

The above problems have different domains but structurally are the same both in premises and solutions. Gick and Holyoak found that presenting the military solution to subjects trying to solve the radiation problem did little to improve performance. Gick and Holyoak suggested that what the subjects needed to develop was a schema of the structure of the latter problem which could be used in the former problem. To develop such a schema, a subject would need at least two structurally similar solutions so that the similarities between the solutions could be discerned.

Gick and Holyoak found that after initially presenting the radiation problem, if two analogous solutions were presented to subjects and the subjects were asked to recall the similarities between the solutions then performance was greatly improved once the subjects returned to the radiation problem.

Such experiments suggest that the structural features of other domains can be picked out by subjects and put to great use in other types of cognition. Such knowledge is of the same level as that represented in story grammars and therefore shows that there is no need to posit a "folklore comprehension" mechanism since similar knowledge is useful in many different tasks and might be stored and processed by a more general mechanism for reasoning in general.

At the core of this issue is a confusion over what constitutes a psychologically interesting grammar.

4.6 When is a grammar psychologically real?

By psychologically real I mean any model which fixes on actual structural or informational features of "in the head" processing. Such a definition is clearly quite loosely defined and broadly speaking has been used in three different ways within cognitive science (Clark [Cla89]:254-255).

The first can be called propositional psychological realism. This is the strongest form of realism. By such a view, a model is psychologically real if the model causally explains A's psychological competence by positing knowledge (possibly unconscious) of the model. The model is internally represented by structures in A's head and there is a one to one match up with every component of the model.

In terms of natural language grammar, such a view necessitates that the grammar is explicitly written in a language of thought. For example, such a view can be found in the work of Chomsky and Katz [Ch+Ka74] or Fodor [Fod80].

The second category can be called structural psychological realism. A model is real if A's competence is causally explained by the fact that A's informational processing capabilities are structured in a way suggested by the model. This view drops any requirement of an explicit language of thought represented model as long as the model fixes onto a functional decomposition of the actual mechanism in the brain. This does not require the model to be unique - for example, a phrase structure grammar, an ATN and a neural network could be psychologically real models of the same psycholinguistic phenomena.

The final view of any cognitive model is descriptive. By this view a good model predicts all and only the external phenomena of the psychological mechanism. For example, a good descriptive grammar would be capable of outputting all and only the well formed sentences of a natural language but make no claims about the actual structure or content of the "in the head" processing needed to generate or comprehend such a sentence.

There has been a conflict between the extreme views of propositional psychological realism and descriptism. Proponents of the former [Fod80] accuse the latter of positing empty and often unfalsifiable theories; whereas those of the latter accuse the former of applying an "engineering viewpoint" to cognitive science (see Devitt and Sterelay [De+St87]:142-146), in other words, expecting the best and most efficient solution to a given problem to be the one employed by cognitive mechanisms - in other words, a cognitive wheel [Den84]. While a certain grammatical formalism might capture very neatly aspects of natural language, it does not follow that our language processing mechanisms evolved in such a neat way. In fact, most of the results of evolution are elaborate kludges which work despite their structure.

Therefore any grammatical formalism intended to model *actual in the head processing* must concern itself not just with the linguistic aspects of language but also with the performance aspects of natural language use in human beings as found in psychological experiments. Most of the grammatical formalisms used, however, have their background in theoretical linguistics and not cognitive psychology and therefore should be regarded as descriptive theories. Such theories cannot justifiably make any direct claims about the internal cognitive mechanisms of language. However, this is not to say that they are of no interest to cognitive psychology since such studies might illuminate general features of cognition [Cla89]:257.

As detailed above, however, story grammars are based firmly in cognitive psychology and do have an impressive corpus of empirical evidence to support them. Where should we place this theory?

Garnham et al assume that story grammars must be read as an attempt at a propositional psychological realistic theory. Such an attempt, they argue fails. It is clear that story grammars cannot make such a claim. It is implausible that humans develop such a specialized mechanism to deal with comprehending a small subset of discourse i.e. simple narratives. Nor, following from the above discussion, is it likely that our comprehension mechanism is nearly so neat as the grammar rules found in story grammars.

However, such an objection can be made to Garnham's and Johnson-Laird's mental models. The strength of mental models lies in their neatness and simplicity - a cause of suspicion for any propositionally real theory. Also, the rules governing mental models and their construction differ from domain to domain. Thus, just as in the case of story grammars, we see posited models of particular types of problem solving - possibly good engineering but also in evolutionary terms unlikely.

Mental models and story grammars are best thought of as examples of structural realism - I will discuss the case only for story grammars.[22]

To ratify the validity of story grammars, we must make certain claims about whatever mechanism processes discourse. Structural features must be used when understanding discourse with structure (e.g folktales). However, given unstructured discourse (i.e. Garnham et al's data), the mechanism must be able to gracefully degrade its performance. This empirical claim has been clearly justified (see above or [Tho77]).

The structure in discourse must be processed in a hierarchical manner and maybe parsed in a way similar to a parse tree. However, this is not to claim that no other linguistic features may affect processing or that recognition of categories is purely syntactic or involves a lexicon.

Nor need it be claimed that the mechanism for comprehending discourse has just this one task. It is plausible that the presence of structural knowledge can be used in other domains - such as problem solving (see above).

4.7 Summary

In the above chapter I have provided a defence for the psychological plausibility of story grammars. I have outlined a range of empirical evidence supporting the claim that such structural information as represented in a grammar is used to

[22] However, Johnson-Laird's work suggests that he views mental models in the same light [Joh83]:462-465, 474-476.

process discourse and defended this theory from two separate critiques.

Most of the evidence reviewed has been based on subjects comprehending discourse rather than on composition. However, it is clear that if such conceptual structure is present in narratives, it must be incorporated during composition - whether or not intentionally by the author. The knowledge of such structure is best represented as a set of grammar rules.

However, the precise nature of the grammar rules and how they match up with parts in an actual story is still problematic. This has been quite rightly pointed out by various critics. I will provide one such mechanism for creating such a link in chapter 6, as part of the implementation of the story generator.

Such deep structure operates at the conceptual level. However, as noted in chapter 2, previous story generators have also failed to deal with the linguistic surface structure of discourse. I will describe such features in the next chapter.

Chapter 5

Features of discourse

5.1 Introduction

Any user of a language can distinguish between a set of isolated sentences and a unified text. It is clear that the whole meaning of a text is greater than its individual sentences [Ha+Ha78]. In fact, individual sentences from a discourse often cannot be interpreted in isolation. For example, consider the following discourse,

A. John and Joan were at home.
B. He liked television.
C. She did as well.

Sentence B's interpretation is dependent on sentence A due to the pronoun "he" referring back to John in sentence A. Furthermore sentence C cannot be interpreted without A and B due to the pronoun reference in A and the anaphoric substitution of "did" for "liked television" from sentence B.

Halliday and Hasan [Ha+Ha78] define such a text as being cohesive - it features linguistic ties which bind the text together. Cohesion is a semantic concept in that it allows the interpretation of propositions in the discourse and defines the sentences as a text - however, as we shall see the mechanisms controlling cohesion can be syntactic, semantic or pragmatic. A cohesive tie presupposes another sentence for its interpretation and often allows the resolution of ambiguity. For example, consider the joke,

A. Time flies.
B. You can't. They're too fast!

[Ha+Ha78]:4

Sentence A can be regarded as complete by itself. However, the humour of the joke depends upon the reinterpretation of A, if B is to be correctly interpretated.

Below, I will outline some of the main features of cohesion and how they can influence the interpretation of discourse.

5.2 Aspects of cohesion

As noted above, various linguistic ties are present in discourse which unify the text into a whole. The types of cohesion can be classified into three main types - reference, substitution and ellipsis, and conjunction. I will discuss each in turn.

5.2.1 Reference

Some entities are referents - they require reference to something else to be interpreted. Therefore, in any text, a referent presupposes another entity to give it meaning. Anaphoric reference presupposes a previous entity. For example,

A, John was a large man.
B, He was very strong.

The pronoun "he" refers back to the previous sentence for its interpretation. Cataphoric referents, conversely, refer to elements which follow - for example,

A, Listen to this.
B, John is retiring.

In A, "this" refers to the proposition contained in B. Both types indicate that information must be looked for elsewhere in the text. However, this could be said of all types of cohesion. Referents are special because the information to be looked for is the actual identity of the referent[1]. In English, there are three types of referent - personals, demonstratives and comparatives. Below is an example of each,

A, Mary was hungry. She missed lunch.
B, I hated Puttletown. I'll never go there again.
C, There were two sweets. However, another was added.

[1] Halliday and Hasan give a nice example of this in the rhyme "Three blind mice. Three blind mice. See how they run. See how they run". In both instances of they, the referent is to the same three blind mice. [Ha+Ha78](page 31).

In A, the personal referent "she" refers to "Mary" in the previous sentence. In B, the demonstrative referent "there" refers to the place "Puttletown". Finally, in C, the comparative referent "another" refers to the class identity "sweets".

5.2.2 Substitution and ellipsis

Substitution and ellipsis both replace linguistic entities in the text. Substitution swaps one entity for another item - for example, in "my fork is dirty. I'll get another one.", "one" is substituted for "fork". Ellipsis is the omission of entities, for example, in the following "Would you like a sweet? I have one more." the entity "sweet" is missing from the second sentence. Substitution and ellipsis are mainly both anaphoric - i.e. they refer back in the text. It is clear that they are also closely related, since ellipsis can be seen as substitution by a null entity. Unlike reference, since both are relations between words (and not concerned with the actual identity of the word), they are syntactic relations.

There are three kinds of substitution, depending on the syntactic category of the substituted term - nominal (nouns), verbal (verbs) and clausal (clauses). Nominal substitution substitutes nouns for typically either "one"[2] or "same". Below is an example of each -

A, All my books are cheap. This one is a bargain.
B, I'll have a bitter. He'll have the same.

In English, verbal substitution involves the verb "do" and substitutes for a

[2] Of course, if the noun is plural then "ones" is substituted.

preceding verbphrase. For example, below "do" substitutes for "drives a car" -

A, Martin drives a car. I do as well.

The final type of substitution is clausal. Here a clause can be substituted by either "so" or "not". A contrastive element always proceeds the substitution. For example,

A, Is it going to rain today?
B, I hope not.
or
B, I hope so.

In both answers "I hope" provides a contrastive environment for the substitution. There are three kinds of contrastive environment possible - report, condition and modality. Below is an example of each,

Report
John said Mary's late. Peter already knew so.
Condition
I hope John's tired. If so, We'll have to go home.
Modal
Will John remember to lock up? He might not.

As noted above, ellipsis can be seen as substitution by zero. For our purposes, I will treat the mechanisms behind ellipsis as being the same as substitution[3]. As in substitution, there are three types of ellipsis - nominal, verbal and clausal. Below is an example of each,

[3]However, this is note quite true - ellipsis operates under a much more complex set of conditions.

Nominal
I saw two police cars. Later, I saw two more.
Verbal
Have you been swimming? Yes I have.
Clausal
Has the plane landed in the airport. Yes it has.

5.2.3 Conjunction

The final kind of cohesion is conjunction. Like reference, conjunction is a semantic relation. However, while reference instructs the reader to look elsewhere for the entity's identity, conjunction instructs the reader that the following sentence is connected somehow with the previous sentence. Typical conjunctive elements are "and", "but"[4], "so" and "then" - all of which are used in different ways. Halliday and Hasan [Ha+Ha78]:76 classify four broad types of conjunction which correspond to the four elements above - additive, adversitive, causal and temporal. Below is an example of each,

Additive
John wanted a car. And more importantly, he wanted a fast car.
Adversitive
John wanted a car. But he had no money.
Causal
John wanted a car. So he went to a car dealer.
Temporal
John wanted a car. Then he wanted a bike.

[4] The conjunctor "yet" could also be used.

5.3 Summary

Above I have briefly described the major features of cohesive text - reference, substitution and ellipsis, and conjunction. However, the overview has concentrated on the aspects of cohesion of particular interest to the project (see [Ha+Ha78] for more detail). All the features operate in a wide range of circumstances and in all types of discourse. However, the actual mechanics of cohesion are still not fully clear. Nor is cohesion an all or nothing affair - a text may be said to be more or less cohesive than another.

The model I have implemented (described in the next chapter) is limited to only exploiting some of the possible situations where cohesive ties are possible. It also presents a more semantically orientated account of cohesion than traditional linguistic accounts. However, the model achieves its main aim of producing readable and cohesive text.

The relationship between the deep structure of discourse (which in this case is governed by a story grammar) and the surface structure (i.e. cohesion) is quite a complex one. The model described in the next chapter, treats these aspects as separate entities and produces stories in several stages - first, the deep structure is created. This is then transformed into an intermediate representation which is then transformed into discourse.

Such a multistage process is hardly psychologically plausible but in the interests of clarity is a practical decision. Nor is there any technical reason why the mechanisms couldn't be combined so that the generation worked in a single pass.

Chapter 6

The implementation

6.1 Introduction

The model below generates simply structured stories as cohesive discourse. Stories are generated in three main phases. First, a story skeleton is created by the story grammar. This skeleton consists of a list of frame like structures called eidons (to borrow Colby's term [Col73]). Then, a microworld is created which allows the story to proceed as specified.

In the second phase, the actual details of the story are fleshed out, using a simple planning system to allow interaction and problem solving by the characters. The planner allows characters to achieve goals specified by the eidons.

In the third phase, this representation is used to generate natural language discourse. A case grammar is used to create an intermediate representation which is then used to generate a cohesive discourse.

Below I will discuss each part of the generation process in more detail and provide some examples of the sort of story generated.

6.2 Story specification

6.2.1 The story grammar

The story grammar implemented most closely resembles a combination of Thorndyke's and Colby's story grammars [Tho77] [Col73]. Like Thorndyke's grammar, there are high level abstract entities such as theme and conclusion. Like Colby's grammar, the terminal entities refer to primary eidons. However, for a practical computational model, certain changes had to made. Some of the primary eidons included by Colby are far too complex to be represented in the program - for example, Colby lists a victory eidon which specifies that the hero wins at a gambling game. To give justice to the subtleties of such an eidon is certainly beyond the scope of the project. Certain eidons were also omitted due to their specific ethnic sensitivity i.e. specific to Eskimo tales, as compared to some elements which are found in all good tales - e.g. a conflict between the hero and the villain. Since the model is not intended to be an anthropological study, I decided to omit such eidons and added eidons which appeared to "fit", including several from Propp's work [Pro68].

The grammar rules are also changed. In particular, the range of categories for terminal entities was expanded from Colby's original grammar to include categories from more abstract grammars. This addition was due to two motivations. Firstly, for practical purposes, certain categories such as setting were useful in keeping track of the story generation. Secondly, such categories appeared to aid the comprehensibility of such tales (see chapter 4 for psychological evidence to support this).

Finally, the most dramatic change was the explicit representation of primary eidons in a frame structure which I will discuss below. The structure specifies the protagonist, his or her goals, other support characters and objects plus a termination condition which states when the eidon finishes. This representation enabled me to specify what information was needed by later parts of the generation program.

The first grammar rule states that any story must consist of a theme followed by a plot and finished by a conclusion.

1, Story $\longrightarrow$ Theme + Plot + Conclusion

The theme is similar in function to Thorndyke's "theme" [Tho77]. However, rather than express explicitly the overall goal of the protagonist, the theme expresses the initial setting and some motivation eidon which gets the story rolling.

2, Theme $\longrightarrow$ Setting + Motivation

The setting is a list of state propositions which describe the world the story is based in. Though such a list might appear artificial in some stories, the setting is a useful way of keeping track of the generator's work. The motivation category can be any of the motivation eidons (see below).

3, Motivation $\longrightarrow$ a eidon from the set of motivation eidons

The conclusion is similar to the setting - it consists of a set of stative propositions which describe the world. However, the conclusion describes the final state of the world at the end of the story.

The plot represents the main bulk of the story and consists of actions and interactions between the characters as they try to achieve their respective goals. The plot consists of a sequence of engagement eidons[1] followed by a resolution eidon.

4, Plot $\longrightarrow$ Engagement* + Resolution

The engagement category can be further split into two categories - an initial set of preliminary action eidons followed by the main action eidon. The preliminary actions are used to build up to the main action which is the major event of the story[2].

Engagement $\longrightarrow$ Preliminary action* + Main action

[1] However, in practice the number of engagements is arbitrarily limited.

[2] In the same manner as the engagement category, these are arbitrarily limited to a small number.

This simple grammar allows the construction of the story skeletons which are used to guide the content of the story.

It should be noted that the main reason for the simplicity of the grammar compared to the grammars discussed above, is due to the omission of the lower categories of such grammars. For example, Rumelhart characterizes an episode (which in some sense is equivalent to an eidon) as being

$$\text{Episode} \longrightarrow \text{Event} + \text{Reaction}$$

Similarly, Thorndyke characterizes an episode as a set of subgoals and attempts followed by an outcome,

$$\text{Episode} \longrightarrow \text{Subgoal} + \text{Attempt}^* + \text{Outcome}$$

However, in the grammar just described such complexities are not needed. Such structures arise implicitly out of the behaviour of the planning system which is used to flesh out the story skeleton produced by the grammar (see below). While such structure is clearly present in stories, it is not necessary to represent it explicitly since it arises naturally out of the behaviour of the simulated agents in the story world. It is interesting to note that according to the empirical data covered above, it is the higher level structure which is most easily recalled. In the model, this corresponds solely to the content of the grammar and not the planning system.

6.2.2 Frame representations of primary eidons

The grammar described above can be used to generate a sequence of eidons which represent the main events of the story to be generated. Other than the setting and conclusion (which are better seen as directives which instruct the generator to describe the world), primary eidons can be represented by the following scheme,

Principal character -
Goal -
Support characters -
Support goals -
Objects -
Termination condition -

I will describe each category in turn. The principal character is the protagonist of the eidon and during planning, is the character who acts first and who the narrative follows. Each eidon might have a different character, causing the scene to shift. In the eidon, the protagonist is marked by a character type - hero, villain etc. During story world creation (see below), the character type is instantiated to an actual character in the story.

The goal is what the protagonist tries to achieve during the eidon. This goal is given to the corresponding character when the eidon begins.

The support characters category lists the type of characters which are needed in the eidon in addition to the protagonist. For example, in a “Villainy” eidon, the protagonist is a villain, and the support must include a victim.

The support characters might have goals themselves. This is often not the case for simplicity. However, sometimes it is useful to supply goals so that more complex interaction can be achieved. For example, in the “Pursuit” eidon, the protagonist, a hero, has the goal of returning home. The supporting villain has the goal of following the hero.

Certain eidons require the presence of objects in the story world. For example, the motivation eidon “Lack” details the search of a hero for some object. The presence of this must be made explicit so that it is included in the story world. The owner of the object can be specified.

Finally, each eidon has a termination eidon which specifies when the eidon can end. Often, this is simply when the protagonist achieves his or her goal. However, the eidon might end when some character enters some state. In practical terms, this condition tells the planning system to stop processing the current eidon and move on to the next part of the story.

As way of an example, consider the following motivation eidon - “Victim deception”

Name - Victim deception
Principal character - villain
Goal - get object-of-desire
Support characters - victim
Support goals - help villain
Objects - owned(victim,object-of-desire)
Termination condition - owned(villain,object)

The eidon requires three entities - a villain, a victim and some object (which the victim owns). The villain is given the goal of acquiring the object and would normally either use deception or force. The eidon finishes once the villain has acquired the object.

6.2.3 World creation

Once the story skeleton is created by the grammar, it is passed to the world creation module which creates a world which can be used to instantiate the story structure. The world initially is a set of empty locations which, during creation various characters and objects can be added to the locations. There are two constraints - firstly, the world created has to contain all the characters and objects specified in the skeleton. Secondly, the world created has to contain only the bare minimum of detail for the story - it cannot have irrelevant entities (which confuse the story) or multiple entities performing the same function (i.e. it only needs one villain etc). The solution to both constraints is to only add entities to the world when necessary - if an entity which already exists can be used by a later eidon, then no new entities should be created.

As an illustration of the process of world creation, consider the simple story skeleton in figure 6.1. The first eidon "Setting" is purely descriptive and therefore requires no additions to the story world. However, the second "Lack" requires a hero and some object of desire. Since neither exist in the world, both are created - the hero is given a name and a set of statistics and the object

Setting

Motivation eidon
Name - Lack
Protagonist - hero
Goal - acquire object-of-desire
Support characters - none
Support goals - none
Objects - object-of-desire
Termination condition - hero starts search for object-of-desire

Main action eidon
Name - donor
Protagonist - donor
Goal - help hero
Support characters - hero
Support goals - none
objects - none
Termination condition - donor satisfies goal

Resolution eidon
Name - Hero satisfaction
Protagonist - hero
Goal - none
Support characters - none
Support goals - none
Objects - none
Termination condition - hero achieves initial goal

Conclusion

Figure 6.1: A Simple story skeleton

chosen from a list of generally desired objects. Both are placed in different locations and replace the tokens "hero" and "object-of-desire".

The third eidon requires a donor as the protagonist and a hero as a supporting character. No donor already exists and so one is created and added to the world (again in a different location to any other entity). Also the donor's name is used to replace the "donor" token. A hero, however, already exists and so the name of the existing hero is used to replace the "hero" token in the current eidon.

The fourth eidon requires no entities which haven't already been created and so the token "hero" is replaced by the hero's name throughout the eidon. Finally, the conclusion eidon is reached - like setting, this is a purely descriptive eidon and therefore requires no additions.

By following this procedure, the program always generates a world which while not having any irrelevant detail, has sufficient content to ensure that the story specified by the skeleton can be generated.

6.3 Story planning

6.3.1 Representation of events

Once the story world has been created, the actual generation can proceed. Story generation involves the updating of two list structures - a temporal list, which represents events in the story in their temporal sequence, and a character list which represents the current state of characters and objects in the world.

Entries in the temporal list are represented in a case grammar based notation for easy translation to natural language (see below). For example, the event of a character called Ivon giving some food to another character Sergi is represented as

> [give, [agentitive(Ivon), dative(Sergi), instrumental(food)]][3]

The temporal list grows as the story progresses as each event generated is added. The character list, however, remains roughly the same size. Instead the contents are updated each time some fact about a particular entity changes. The character list contains the statistics of both characters and objects - for example, it keeps track of the location, heath, mental state and knowledge of every character in the world. For example, below is a typical entry for a character

> [name(Ivon), sex(male), at-location(hill), mental-state(10),
> Heath-state(10), knows(at-location(Sergi,forest))]

Actors can learn about the world and make inferences. The resultant knowledge is represented by a statement that the character knows some particular fact. Though not a very sophisticated scheme, this is sufficient for the types of knowledge and inference which the characters can discover. Though such knowledge can be represented by this scheme, characters cannot gain any meta-

[3] The classes agentitive, dative and instrumental correspond to categories in the case grammar (see below).

knowledge - for example, that another character knows some fact. This avoids some of the problems which Talespin encounters when characters try to second guess each other. However, it also simplifies the possible interactions between different characters - which at this stage is not a problem anyway, but future improvements to the program must deal with these knowledge representation problems.

6.3.2 The planning system

The story is generated by processing each eidon in sequence. The current eidon is inspected and the goals of the protagonist added to the corresponding character. Control is then given to the planning system. Once the termination condition of the eidon is satisfied, the next eidon is processed until all have been processed.

There are two exceptions - as noted above, “Setting” and “Conclusion” eidons are both purely descriptive. Instead of passing control to the planning system, the eidons instruct the program to add to the temporal list a description of the state of all the characters in the world.

The planning system is used to simulate the behaviour of the characters associated with each eidon. The system works in a loop which stops only if either the terminal condition of the current eidon is reached or the protagonist (as specified by the eidon) cannot make any more progress with the current goal.

Characters achieve their goals as follows. Firstly, the character's knowledge of the world is updated (i.e. inferences are made by the character about the previous events - see below for further discussion). If the goal is a general one then a set of problem strategies are used to add subgoals to the character which must be solved before the main goal is achieved. The last subgoal added is the first goal which must be achieved (see below for a discussion of problem solving strategies).

If the goal is more particular to a given situation and therefore not covered by the strategies then a form of means - ends analysis is used to plan a set of actions to achieve the goal. The planner looks for a basic action (see below) which would achieve the current goal. The preconditions of the action are inspected and any which aren't currently true are added as subgoals which must be achieved first. For example, an action might need to be performed by another character, in which case the subgoal of persuading the other character to perform the action would be added to the character's record.

An action which achieves the current goal immediately and whose preconditions are all true is carried out by the character immediately. Once an action has been performed by the character, control of the planning system is passed onto another character at the same location - thus allowing interaction between characters.

Since each eidon specifies a protagonist, control of the planning system is first given to this character. Once the protagonist has acted, control is passed to any other character present. This allows each character to make one action each time they have control of the planning system. Note that characters not in the same location as the protagonist can't act since control of the planning system isn't ever passed to them. This makes the system a poor simulation of the real world but provides two advantages. Firstly, the scheduling between characters and eidons is much more simple and secondly, and more importantly, no action is performed without being output to the temporal list. This ensures that any relevant action to the story is mentioned in the generated story.

6.3.3 Inferences

Like Talespin, the story generator has to make inferences. Inferences are not only needed to access the effects of events but also for characters to plan solutions for goals by specifying the preconditions of the goal as subgoals. For example, the precondition for interacting with another character is being in the same location. Any character who has such a goal but is not in the same location is given this subgoal (see below for a further discussion on problem solving strategies).

However, the consequences of events constitute the majority of inferences produced. The consequences of the previous set of events are calculated before the planning of any character since inferences can change the behaviour of the character by altering their health or mental states or relationship between other characters. Some of the inferences depend on the current state of the world and the individual knowledge of each character - for example, a character might make different inferences depending whether he or she likes or hates the other character.

The inferences made by characters present at an event are calculated using a set of production rules. Details of the event are matched to these rules which specify the inference made. For example, consider the action of “Sergi gave Ivon some food”, this would be represented as

[give [agentitive(sergi) dative(ivon) instrumental(food)]]

The production rules specify that any character giving something to another character must like that character. Therefore, Ivon infers that Sergi likes him. Conversely, any character receiving a gift will be grateful, Ivon also decides that he likes Sergi. Sergi will also infer that Ivon now likes him - a precondition for persuading Ivon to do some task.

This system isn’t very sophisticated but allows a reasonable amount of interaction to make the stories generated interesting. For example, as above it allows characters to use bribes to persuade other characters to perform actions.

6.3.4 Actions

Within the world characters are allowed to perform a number of basic actions to achieve their own goals. Most of the actions have preconditions - for example, to "take" an object, the character must be in the same location and the object mustn't be owned by any other character. As noted above, actions are represented in the temporal list in terms of a case grammar. Therefore, each action is represented in the program in terms of its case grammar representation, preconditions and effects. For example, the action "give" is represented as -

Action - Give.	
Form - [Give, [character(A), dative(D) instrumental(O)]]	
Preconditions -	A + D are at the same location
	A owns O
Effect -	A stops owning O
	D owns O

In total there are twelve different actions, covering the acquiring of objects (give and take), movement (move, hide, dance), the acquiring and communication of knowledge (learn and tell), character interaction (bargain, attack, deceive) and decision making (agree and refuse). Such a group of actions appears to be the minimum number required for reasonably sophisticated behaviour.

6.3.5 Problem solving strategies

To achieve common goals, there are a set of problem solving strategies which specify possible approaches and heuristics which can be used by the characters. Some are generic to all problem solving - for example, there is a rule which states that to affect another entity, the character must be in the same location.

Other strategies are more specific. For example, "trans", a strategy for moving to objects and locations, first plans the shortest route to the location (if the location is known, otherwise the strategy "aqknow" for acquiring knowledge is invoked) is planned and a subgoal added of moving to the first intermediate location. However, if the target location is adjacent then the character simply "moves" to the new location.

Finally, some strategies are eidon and character specific. For example, there is a strategy for satisfying villainy eidons - this includes subgoals such as harming other characters and stealing objects.

6.3.6 Plot devices

As noted above, each eidon is processed till its termination condition is satisfied. This typically is connected to the protagonist achieving some goal or entering some state. However, often the characters can get stuck so that no more progress can be made. Such problems stop the story from progressing and therefore have to be avoided.

To ensure this, the program has a plot device module. Plot devices are essentially tricks and set pieces which facilitate the continuation of the story[4]. Below I will give some examples of the plot devices used in the system.

One of the most common problems for characters is lack of knowledge - a

[4]Therefore plot devices are extremely close to Colby's concept of secondary eidons which, as described in chapter three, are used to ensure that the story runs smoothly [Col73].

character might have a goal to get some object or interact with some other character but mightn't know where to go. Therefore, the subgoal of acquiring this knowledge is then added which invokes the "aqknow" problem solving procedure (see above). "Aqknow" involves finding another character and asking them. However, if the character doesn't know where any other characters are then they are stuck. This would halt the problem solving module and pass control to the plot device module. Typically when "aqknow" type problems are unsatisfiable, a small animal is dispatched with the knowledge required. The character can then gain the required knowledge and the story can continue. An example of this occurred in the following story,

> nikita was at the hill. he wanted to get some food. so he wanted to know where some food was at. but he couldn't make any progress. A magpie came to the hill. he knew nikita was there. nikita knew the magpie was. he wanted to persuade the magpie to tell him where some was at. the magpie bargained nikita that if nikita danced then it would. nikita agreed to dance. and the magpie agreed to tell nikita where some was at. nikita danced. he was grinning. the magpie knew this. he told nikita that some was at the beach. nikita liked the magpie. he walked to the beach. the food was there. nikita took it. and he was satisfied.

Other plot devices are used to ensure that future eidons are possible - for example, if the next eidon features a victory of the hero over the villain but the villain is stronger then a plot device ensures that the hero is given some advantage - e.g. given a magic weapon etc.

Another problem occurs if a character dies prematurely i.e. dies before being needed in a eidon. Propp [Pro68] notes that when villains die, the role is immediately taken over by a female relative. In a similar vein, when a character prematurely dies, the program generates a sibling to take over the character's role.

6.3.7 Character predispositions

Each eidon requires specific character types. As discussed above, the variety and function of different character types was discussed by Propp [Pro68]. In a similar vein, the program distinguishes between five different character types - hero, villain, victim, donor and false hero[5]. Each character type performs different roles in each eidon (which of course, depends on which goals are assigned to each character).

Different character types also have predefined dispositions to interaction between other characters. As noted above, Talespin [Mee76] requires the user to specify that different characters like or trust each other. This is not necessary since such details are predefined for each character type. For example, heros and victims are both honest and like all other characters (e.g. they will agree to any bargaining). Donors are neutral but always honest. False heros are always dishonest but will agree to any bargain, only to break their promise. Finally, villains are dishonest and antagonistic to any other character.

[5] As discussed in chapter 3, Propp lists 7 different character types, however to generalise the stories created I have omitted the "dispatcher" and "helper" characters and renamed the "princess", "victim".

Such stereotypes are rather crude. However, they allow the program to make sensible choices as to the behaviour of characters. Such predispositions are also modifiable - characters can change their opinions due to another character's behaviour.

6.4 Linguistic generation

The system generates natural language by translating the case grammar based propositions into natural language text. To do this, the case grammar translates the propositions into an intermediate representation which is then processed into text by the discourse processing module. I will first discuss the case grammar and how it generates the intermediate representation. I will then show how this representation can be manipulated into a cohesive text.

6.4.1 Case grammar

Case grammar was originally developed by Fillmore [Fil68] as a contrast to Chomsky's syntactically orientated transformational grammar [Cho57]. It describes language from a semantic perspective and describes the functional relationships between phrases in sentences. This has the advantage of being able to reflect the deep structure similarities between syntactically different but conceptually similar propositions. For example, the first set of sentences below are conceptually the same despite having different syntactic structures. The second set, however, have similar surface structures yet refer to quite different events[6].

1, I opened the door.
The door was opened by me.
2, I baked the cake.
The bread baked to a cinder.
The oven bakes evenly.

English primerly uses each constituent's place in the sentence to express its role in the sentence. As can be seen from the above examples, this can often cause ambiguities to arise. However, certain other languages use explicit markings, e.g. morphological affixes, to indicate role. For example, in Russian, "professor uchenika tseloval" means "the professor kissed the student" while "professora uchenik tseloval" means "the student kissed the professor"[7]. Russian provides six possible cases - nominative, genitive, dative, accusative, instrumental and propositional and a corresponding set of suffixes which attach to nouns specifying their role which reflect the deep structure of Russian explictly in the surface form.

Case grammar argues that such a system operates in English, not like Russian in the surface form, but in a sentence's deep structure[8]. The grammar focusses on the main verb of the sentence and each noun in the sentence is given a case which reflects its role in the sentence.

[6]It could be argued that such distinctions are purely semantic. However as Winograd [Win83]:313 notes, traditional linguists have used case grammar concepts to resolve other more clearly syntactic issues - for example, see Stockwell, Schacter and Partree's UCLA English grammar [St+Sc+Pa73].

[7]This example is from [Win83].

[8]Of course, there is a limited case system in the interpretation of pronouns in terms of their number and gender.

Agentative - animate instigator of action.
Instrumental - inanimate force involved.
Dative - animate recipient of action.
Factitive - object resulting from action.
Locative - location.
Objective - anything else.

Figure 6.2: Fillmore's original set of cases for English [Fil68]

A number of case systems for English have been proposed. The original and best known being Fillmore's six cases - agentative, instrumental, dative, factitive, locative and objective (see figure 6.2) which is representative of later case systems.

For example, Simmons and Celce [Sim73] used the following set of cases in a computer program to generate English sentences - Causal actant, Theme, Locus, Source and Goal. In fact, case grammar bears a strong resemblance to Schank's more semantically orientated conceptual dependencies except in conceptual dependency theory, the "cases" are filled with conceptual rather than syntactic entities.

As the above shows, it is still largely undecided what cases a case grammar should include. In fact, one possible criticism of case grammar is the suggestion that the number of cases needed in such a system might be infinite. There is also some controversy over what case should be allocated to certain constituents. For example, in 1 below, whether "the wind" be classed as "agentitive" (as in 2) or "instrumental" (as in 3).

1, The wind opened the door.

Agentitive - animate actor of action.
Instrumental - object involved.
Dative - animate recipient of action.
Propositional - an embedded proposition.
Locative(initial) - initial location.
" (final) - final location.
Conditional - a condition of the action.
State - a state which an object can be in.

Figure 6.3: The case grammar used in the story generator

2, He opened the door.
3, The key opened the door.

6.4.2 Intermediate generation

Despite the difficulties noted above, case grammar is a very practical and flexible representation for generation due to its strong semantic orientation. The grammar I implemented uses the cases in figure 6.3. The choice of these cases was motivated mainly for the practical requirements of the program rather than for any linguistic theory.

Each verb in the lexicon has an associated case frame. This specifies which case constituents are necessary, optional or forbidden to occur with the verb. For example, below is the case frame for "to give". Accompanying the proposition, is a mode list which specifies the following features - tense, interogative (i.e. is the proposition a question) and whether the prosition is negative.

Verb - to give.

Necessary - Agentitive, Instrumental.
Optional - Dative, Locative(initial).
Forbidden - Locative(final), Conditional, Propositional.

Generation is accomplished by placing the case constituents in their correct positions relative to the main verb and embedded verbs. Prepositions are added by the generator to the front of certain cases - for example, "on" is added in front of the locative in the following example,

Initial proposition
Verb - to be.
Modelist - tense(past)
Instrumental - food.
Locative(initial) - table.
↓
Intermediate representation
Instrumental(food) was on locative(initial)(table).

Negations are accomplished by inserting "don't", "didn't" or "won't", depending on tense, before the main verb, and interogatives (which are always embedded in another proposition - for example, "Sergi asked Ivan where the food was") are formed by prefixing the sentence with an appropiate interogative. For example, consider the following proposition,

Verb - to know
Modelist - tense(past), negative.
Agentitive - Sergi.

Propositional - | Verb - to be. |
| Modelist - tense(past), interogative. |
| Instrumental - food. |
| Locative(initial)(-) |

The embedded proposition is dealt with first. The cases are placed in their correct positions around the verb.

Instrumental(food) Verb(be) Locative(initial)(-)

Since the mode list specifies an interogative, "where" is attached to the front of the proposition due to the presence of the locative case. The mode list also specifies past tense, and so the conjugation of the verb "to be" is looked up and used. The preposition "at" is placed in front of the locative. Since the locative contains a null value, it is dropped. This gives -

where Instrumental(food) was at

The top level proposition can then be dealt with. As before, the cases are placed in their correct positions.

Agentitive(Sergi) verb(know) Prop(where Instrumental(food) was at)

Since "negative" appears in the mode list, "didn't" is added, in front of the verb and the present tense is used. Giving the intermediate representation -

Agentitive(Sergi) didn't know Prop(where Instrumental(food) was at)

This representation is then passed to the discourse processing module for the construction of cohesive ties to be built between it and the rest of the discourse.

6.4.3 Discourse generation

As noted above, the case grammar generates an intermediate representation which can be used to process the narrative into a cohesive text. Figure 6.4 is a typical excerpt of such a story.

As noted above in chapter 5, discourse can be recognised by the presence of cohesive features - e.g. reference, substitution and ellipsis, and conjunction. Any model of discourse must be able to handle these features. However, the needs of generation versus interpretation are, as we shall see, quite different.

The most common technique used in discourse processing is focussing [Gro77]. Focussing involves keeping track of (or focussing on) the current entities and topics in the discourse[9]. Such focussing allows the disambiguation of sentence meanings. As Grosz notes, though a word might have several different uses, its immediate context (and the topic of the discourse) typically constrain it to one meaning.

Such focussing often uses a history list or stack. For example, Sidner [Sid79] uses the following parameters to handle the interpretation of anaphoric referents such as pronouns - the current focus (i.e. referent), potential constituents in the sentence for a change in focus, and past foci in the text (held in a stack). Constituents are added to the list when they occur in the current sentence. The current focus is added to the stack when the focus changes. Previous foci are

[9] Though I will just consider text, the use of focussing has also extensively been used in dialogue.

1, [agentitive(nikita) wanted
proposition(agentitive(nikita) to own instrumental(sword)]
2, [agentitive(wilma) owned instrumental(sword)]
3, [agentitive(nikita) wanted
proposition(agentitive(nikita) to persuade dative(wilma)
proposition(agentitive(wilma) to give dative(nikita)
instrumental(sword)))]
4, [agentitive(nikita) asked dative(wilma)
proposition(agentitive(wilma) to give dative(nikita)
instrumental(sword))]
5, [agentitive(wilma) bargained dative(nikita)
conditional(if agentitive(nikita) told dative(wilma)
proposition(where instrumental(food) was at)
then
proposition(agentitive(wilma) would give
dative(nikita) instrumental(sword))]
6, [agentitive(nikita) didn't know
proposition(where instrumental(food) was at)]
7, [agentitive(nikita) refused dative(wilma)]
8, [agentitive(wilma) was state(angry)]
9, [agentitive(wilma) attacked dative(nikita)]
10,[agentitive(nikita) was state(angry)]
11,[agentitive(nikita) hated dative(wilma)]
12,[agentitive(nikita) attacked dative(wilma)]
13,[agentitive(wilma) was state(dead)]
14,[agentitive(nikita) knew proposition(agentitive(wilma) was state(dead))]
15,[agentitive(nikita) took instrumental(sword)]
16,[agentitive(nikita) was state(satisfied)]

Figure 6.4: Excerpt of an narrative in its intermediate form

popped off the stack and used, when the discourse returns to previous topics.

For interpretation, such focussing mechanisms have to be quite complex since the anaphor needn't be directly after the object referred to. Below is a classic example, due to Grosz and cited by Sidner [Sid83]:315.

> A: One of the bolts is stuck and I'm trying to use both the pliers and the wrench to get it unstuck.
> E:Don't use the pliers. Show me what you're doing. Show me the 1/2" combination wrench.
> A:OK.
> E:Show me the 1/2" box wrench.
> A:I already got it loosened.

Notice that the "it" in the last sentence refers back to the "bolts" in the first sentence, rather than the most recent object (i.e. the "box wrench") mentioned. A model of discourse interpretation must deal with such difficult examples. However, this is not to say that such examples are common. Hobbs [Hob78] statistically studied one hundred examples of pronoun use in three different forms of text. He found that ninety eight percent referred to either an entity in the current sentence or the immediatly proceeding sentence. Nor should such a convoluted use of anaphora be desirable. As reviewed above, the extra work entailed by disambiguation slows reading time and lowers comprehension of the text[10].

Therefore, any discourse generation program should generate easy to read text. Cohesion must be present but it must be used to aid comprehension and

[10] For a more detailed discussion see [Joh83]:356-396.

not cause more effort on the reader's part. Below I outline the case based approach used here.

6.4.4 A case based approach to discourse processing

The program handles cohesion by processing each proposition in sequence[11] and maintaining a register of the last instance of each case and sentence constituent (e.g. the main verb) in each proposition. It produces text with anaphoic cohesion (see chapter 5 or [Ha+Ha78]).

The register is updated for every unembedded instance of either a case or sentence constituent. This simple method can be used to create cohesive features. I will discuss its use in further detail with respect to each feature.

Reference

Noun phrases are replaced by pronouns when they match the current contents of the register of their allocated case. For example, if a dative in the proposition matches the last instance of a dative (i.e. the entry in the register) then it is replaced by a pronoun. Which pronoun is used depends on the gender and number of the noun phrase.

Some intra sentence reference might also be needed. For example[12],

[11] Note that this means the program only generates anaphoric and not cataphoric phrases. However, anaphora represents the majority of cohesion.
[12] In the examples which follow, numbered examples refer to statements in figure 6.4.

4, [agentitive(nikita) asked dative(wilma)
proposition(agent(wilma) to give dative(nikita) instrument(sword))]

Despite the two instances of "nikita" being in different case categories, the second use must be replaced by the accusative pronoun "him", giving[13] -

"nikita asked wilma to give *him* the sword"

Definite versus indefinite articles

A nounphrase constituent which doesn't match the entry is transformed into a full noun phrase. With the exception of proper nouns, this involves adding a determiner in front of the constituent. The program keeps a list of objects which are mentioned in the discourse. Any new object mentioned is added to this list (with the exception of objects in goal propositions). If the object is being mentioned for the first time, an indefinite determiner is added to the nounphrase - for example, either "a" or "some" depending on whether the object is a count or mass noun. Objects which have already been mentioned are given the definite determiner "the".

Substitution and ellipsis

Substitution and ellipsis are handled in much the same way as reference. Constituents in the proposition are inspected. If they match the previous instances

[13] Note that the second instance of "wilma" is also dropped, see below.

held in the register then they are either replaced by a substitution or removed by ellipsis.

Objects are replaced or ellipted if they match the last instance of their category but refer to a different entity in the story world [14] - for example, two magpies might be present. Objects are substituted by "one" if the sentence is in the past tense and "same" if in the future or present. Substitution occurs if the case category of the noun is essential to the verb (as recorded in the lexicon). If the noun is optional, then it is ellipsed.

If the verb and following constituents (i.e. the verb phrase) match the last instance of a verb phrase then it can be substituted (by "do") or ellipted. The choice depends on whether the substitution keeps the sentence grammatically correct. If so, substitution is preferred, otherwise ellipsis is used (and the verb phrase is simply omitted).

There is also the possibility of just the main verb being ellipted if the surrounding case nouns differ between the sentences. For example,

a, agentitive(nikita) took instrumental(sword).
b, agentitive(wilma) took instrumental(food).
↓
a, nikita took the sword.
b, wilma the food.

Finally, propositions or clauses can also be changed. As above, if the current

[14]Note, if the noun phrases refer to the same object, then it is a case of reference and not substitution - see above.

proposition or conditional matches the last recorded instance (which could be the entire previous sentence, in the case of a proposition) then it can be replaced. If a contrastive environment is available, then the proposition is replaced by "so" or "not" depending on whether the environment is positive or negative. If no environment is present, then the proposition is ellipted.

Though not a cohesive feature, some ellipsis is used within each proposition. For example,

> 1, [agentitive(nikita) wanted
> proposition(agentitive(nikita) to own instrumental(sword))]

In such a case, the agent of the embedded proposition must be dropped.

Conjunction

As noted in chapter 5, there are four types of conjunction - additive, adversative, causal and temporal. Each is used when the program recognises a common relationship between adjacent sentences.

Additive conjunction occurs when a pair of statements have the same subject. For example[15] -

> 10, [agentitive(nikita) was state(angry)]
> 11, [agentitive(nikita) hated dative(wilma)]
> 12, [agentitive(nikita) attacked dative(wilma)]
> ↓

[15]Note that conjunction operates between *pairs* of sentences and so proposition 12 isn't cojoined.

10, nikita was angry.
11, and he hated wilma.
12, he attacked her.

Adversative conjunction occurs when a goal is stated but the next statement blocks the direct satisfaction of that goal. For example,

5, [agentitive(wilma) bargained dative(nikita)
conditional(if agentitive(nikita) told dative(wilma)
proposition(where instrumental(food) was at) then
proposition(agentitive(wilma) would give
dative(nikita) instrumental(sword))]
6, [agentitive(nikita) didn't know
proposition(where instrumental(food) was at)]
↓
5, wilma bargained nikita if nikita told her where some food was at then
she would give nikita the sword.
6, but nikita didn't know where the food was at.

Causal conjunction occurs when the proceeding event causes a character to make some inference which modifies his or her behaviour. For example,

7, [agentitive(nikita) refused dative(wilma)]
8, [agentitive(wilma) was state(angry)]
↓
7, nikita refused wilma.
8, so wilma was angry.

Finally, temporal conjunction occurs when the first proposition refers to some physical action and the second is caused by the completion of that action and has the same subject. This ensures that the conjunctor "then" refers to the event temporally following the action. For example,

15, [agentitive(nikita) took instrumental(sword)]
16, [agentitive(nikita) was state(satisfied)]

1, nikita wanted to own a sword.
2, wilma owned one.
3, so, nikita wanted to persuade wilma to give him the sword.
4, he asked her this.
5, wilma bargained nikita that if nikita told her where some food was at then she would give him it.
6, but nikita didn't know this.
7, so, he refused wilma.
8, so, wilma was angry.
9, she attacked nikita.
10,so nikita was.
11,and he hated wilma.
12,he attacked her.
13,wilma was dead.
14,nikita knew this.
15,he took it.
16,then he was satisfied.

Figure 6.5: Excerpt of a narrative in its natural language form

↓
15, nikita took the sword
16, then nikita was satisfied

The following methods can be used to transform a set of propositions such as those in figure 6.4 into a piece of natural language text such as that in figure 6.5.

6.5 Examples of generated stories

Below are three example stories which have been generated by the model described above.

nikita was at the mountain. sergi was there. nikita was brother to sergi. sergi to nikita. nikita was good. sergi bad. nikita wanted some gold. he wanted to know where some was at. so he searched for some. he walked to the hill. but he couldn't make any progress. a wolf came there. the wolf knew nikita was there. nikita the wolf was. he wanted to persuade the wolf to tell him where some was at. the wolf told nikita to jump. nikita was scared. so nikita agreed. he did. the wolf grinned. he told nikita that some gold was at the cave. nikita liked the wolf. he walked to the cave. some was there. he took it. and he was satisfied. sergi wanted some gold. so he wanted to know where some was at. he searched for some. he walked to the hill. the wolf was there. sergi wanted to persuade the wolf to tell him were some food was at. the wolf told sergi to jump. sergi was angry. he attacked the wolf. the wolf was too. it attacked sergi. sergi was dead. the wolf knew this. and it was satisfied. nikita was satisfied. sergi was dead. The end.

wilma was at the hill. she was good. she had some food. nikita was at the cave. he was bad. Ivon was at the forest. he was good. nikita wanted to do some villainy. he walked to the hill. he knew that wilma had some food. he wanted to own it. he bargained wilma that if wilma gave him the food then he would give wilma some gold. so wilma wanted some gold. she agreed to give the food to nikita. she did. nikita attacked wilma. wilma was injured. nikita walked to the cave. ivon walked to the hill. he knew that wilma was injured. so he was angry. wilma told ivon that nikita had taken the food. so ivon was angry. he hated nikita. ivon walked to the cave. he attacked nikita. nikita was injured, he attacked ivon. ivon was. he attacked nikita. nikita was dead. ivon knew this. he took it. then he walked to the hill. he gave it to wilma. and he was satisfied. wilma was. nikita was dead. The end.

sergi was at the forest. he was good. wilma was at the cave. she was bad. she was a dragon. she had some gold. sergi wanted to roam. he walked to the cliffs. a old man was there. the old man asked sergi to give some food to him. so sergi agreed to give some to the old man. but he didn't have some. so he wanted to get some. but he didn't know where some was at. he asked the old man to tell him. the old man told sergi that some was at the forest. sergi walked to the forest. some was there. he took it. then he walked back to back to the cliffs. he gave it to the old man. the old man was happy. sergi knew this. the old man gave a magic sword to sergi. so sergi was happy. he liked the old man. sergi walked to the cave. wilma the dragon knew sergi had the sword. sergi that wilma had the gold. she told sergi to give it to her. sergi refused wilma. wilma attacked sergi. sergi was very injured. and he was angry. he hated wilma. he attacked wilma with it. wilma was dead. sergi took the gold. he was happy. wilma was dead. The end.

The stories are all well structured and are (for their genre) quite plausible.

The linguistic generation isn't bad either. However, it is clear that at times, the cohesion goes too far. For example, in the second story, the pronoun "he" is used instead of Ivon too many times to be pleasing to read. However in most cases, the cohesion helps the readability of each story.

Chapter 7

Conclusion

This dissertation has described the development and implementation of a story generating model. The model has drawn from a wide number of sources - anthropology, cognitive psychology, linguistics and artificial intelligence.

The model is capable of generating well structured stories of varying complexity. The output text is quite readable and displays cohesive features. The work has some quite important implications which I will discuss below.

The model generates stories, which I believe, are at least better structured and more interesting than those generated by previous attempts such as Meehan's Talespin. More importantly, the model attempts to cover all aspects of story composition. That is, the stories are generated without any user involvement. Unlike some previous attempts, the program decides the setting and shape of the story before generation rather than rely on the user or random luck to ensure an interesting story subject. Therefore, the difficult problem of representing "metaknowledge" about stories has been tackled and in some sense the program knows what makes a story good.

Regardless of the quality of the stories generated, the model provides a useful formalism for representing story knowledge. The abstract structural units of the story are represented in a grammar and frame type data structures (eidons) while the actual content of each episode in the story is expressed in terms of steps in a planning system. This dualistic representation allows a finite lexicon while also allowing for the unpredictable and original nature of stories.

This dissertation has solely concentrated on the generation of stories. However, I feel confident that such a dual mechanism could also be employed in story comprehension with a planning system used to classify parts of the story into lexical categories for the grammar which could be used to parse the structure of the tale and make predictions of what the next category would be.

One unexpected feature of the story generation which was quite pleasing was the merging and omission of eidons. As discussed above, the grammar specifies the story in terms of a set of eidons in a story skeleton. Each eidon is processed in turn by the planning system to generate the details of the story. The program passes to the next eidon when the termination condition of the current eidon is true. However, often the condition is true as a side effect of previous events in the story. In this case the eidon isn't processed at all and omitted. Thus, a complex story skeleton might shrink to a more compact story. I didn't anticipate this but this effect is discussed with regard to human folktales by Propp [Pro68].

During the dissertation I have presented a defence to the numerous critiques of story grammars within cognitive science. I have argued that such grammars are useful descriptive tools of the types of knowledge and structure which are used in comprehension and composition of discourse. This is not to say that such grammars represent all the knowledge and processes involved, however, and I have shown how such grammars can be combined with other cognitive models proposed by different theorists. The cognition story grammars describe is, I believe, not limited just to discourse processing. Rather, such knowledge use appears central to other problem solving areas where structural or syntactic knowledge is important - for example, reasoning by analogy.

The one objection which I omitted defending against is that such grammars are non-computable, abstract and imprecise. It is telling that no psychologically motivated story grammar has been implemented on a computer. Rather, researchers have used their own intuitions as to what part of a particular story matched which category of the grammar. This has, unfortunately, resulted in a wide variety of different grammars with quite different representational powers. It has also allowed the above objection some force.

However, the implementation of my program stands as a direct counter-example to such a claim. I have shown not only that such grammars are psychologically plausible but also how one might be implemented computationally.

The grammar implemented was rather ad hoc and drew from several different existing grammars (most notably, Colby's grammar [Col73] and Thorndyke's [Tho77]). However, the model provides a possible future basis for comparing the ability of different grammars - by intepretting the grammars objectively rather than using human intuition.

The prototype generates quite readable discourse. However, only a limited range of linguistic features were incorporated - for example, the program cannot generate cataphoric forms of cohesion, only the more common anaphora. This was due partly to lack of time but also due to the use of case grammar. Traditional linguistics has mainly ignored work in case grammar and such discourse features as ellipsis haven't been discribed in such terms. Therefore, the program was not able to capture the full subtlety of cohesive features. However, case grammar is a very practical formalism and very attractive from an A.I. point of view due to its strong semantic flavour. I believe that further progress can be made using case grammar to formalise such aspects of discourse. The generator was also able to generate text quickly - this contrasts with more traditional discourse generation models, such as McKeown's TEXT system [McK85], which are slow in comparision. Therefore, case grammar is a useful formalism for natural language generation where speed is important. For example, such a technique would be useful in interactive dialogue generation.

Since story generation is a difficult task, the prototype can be expanded in almost any direction. Probably the most important direction is the planning system. At present, the planning system uses a rather crude form of means-ends analysis. This is sufficient to allow the characters to behave intelligently and solve problems. However, improvements can be made by using a more sophisticated planning system. For example, a greater depth of behaviour could be created if the characters where able to solve problems with multiple constraints.

To conclude, I have presented a prototype which generates well structured stories as specified by a grammar. The stories generated incorporate some of the aspects of cohesive text and on the whole are quite readable. Future directions include enhancing the linguitic generation and allowing for more sophisticated planning.

Bibliography

[An+Sl90] Anderson S. and Slator B.M. (1990) Requiem for a theory: the story grammar story. In *Journal of Experimental and Theoretical Artificial intelligence*, **2**,3:253-275.

[Bar32] Bartlett F.C. (1932) *Remembering.* Cambridge, UK: Cambridge University Press.

[Bet76] Bettelheim B. (1976) *The uses of enchantment.* UK:Thames and Hudson.

[Bl+Bo80] Black J.B. and Bower G.H. (1980) Story understanding as problem-solving. *Poetics* **9**: 223-250.

[Bl+Wi79] Black J.B. and Wilensky R. (1979) An evaluation of story grammars. *Cognitive Science* **3**: 213-230.

[Bod77] Boden M.A. (1977) *Artificial intelligence and natural man.* New York:Basic Books.

[Bod91] Boden M.A. (1991) *The creative mind : myths and mechanisms.* New York:Basic Books.

[Br+Ne78] Bruce B. and Newman D. (1978) Interacting plans. *Cognitive science*,**2**:195-233.

[Ch+Ka74] Chomsky N. and Katz J. (1974) What the linguist is talking about. In N. Block (ed) *Readings in the philosophy of psychology* (Vol2). London:Methuen and Co (1980).

[Cho57] Chomsky N. (1957) *Syntactic structures.* The Hague:Mouton.

[Cho65] Chomsky N. (1965) *Aspects of The Theory of Syntax* Cambridge, MA: MIT Press.

[Cla89] Clark A. (1989) *Microcognition.* Cambridge, Mass: Bradford/MIT press.

[Cla75] Clark H.H. (1975) Bridging. In R.C. Schank and B.L. Nash-Webber (eds) *Theoretical issues in natural language processing: preprints of a conference at MIT*. Mass:MIT press.

[Coh81] Cohen H. (1981) *On the modelling of creative behaviour*. Santa Monica, California:Rand Corporation. Rand paper p. 6681.

[Col73] Colby B.N. (1973) A partial grammar of Eskimo folktales. *American Anthropologist*: 645-662.

[Co+Qu72] Collins A.M. and Quillian M.R. (1972) Expriments on semantic memory and language comprehension. In L.W. Gregg (ed) *Cognition in learning and memory*. New York :Wiley.

[Cul76] Culicover P.W. (1976) *Syntax*. New York: Academic press.

[DeB+Co79] De Beaugrande R. and Colby B.N. (1979) Narrative models of action and interaction. *Cognitive science* Vol **3** No. **1**: 43 - 66.

[Deh81] Dehn N. (1981) Story generation after Talespin. *Proceedings of the 7th international conference on artificial intelligence*,16-18.

[Den84] Dennett D.C. (1984) Cognitive wheels: The frame problem of A.I.. In C. Hookway (ed) *Minds, machines and evolution*. Cambridge: Cambridge University Press.

[De+St87] Devitt M. and Sterelny K. (1987) *Language and reality*. Oxford:Blackwell/MIT press.

[Dun45] Duncker K. (1945) On problem solving. *Psychological Monographs*,**58**:1-113. (Whole No. 270).

[Dun65] Dundes A. (1965) On computers and folktales. *Western folklore* **24**: 185-189.

[Dye83] Dyer M. (1982) *In depth understanding*. MIT press, Cambridge MA:MIT press.

[Eh+Jo82] Ehrich K. and Johnson-Laird P.N. (1982) Spatial descriptions and referential continuity. *Journal of Verbal learning and Verbal behavior*,**21**:296-306.

[Fil68] Fillmore, C.J. (1968) The case for case. In E. Bach and R.T. Harms (eds), *Universals in linguistic theory*. New York, Holt, Rinehart and Winston.

[Fi+Ni71] Fikes R. and Nilsson N.J. (1971) STRIPS: A new approach to the application of theorem proving in problem solving. *Artificial intelligence*,**2**:189-208.

[Fod80] Fodor J. (1980) Some notes on what linguistics is about. In N. Block (ed) *Readings in the philosophy of psychology (Vol2)*. London:Methuen and Co.

[Ga+Me89] Gazdar G. and Mellish C. (1989) *Natural language processing in Prolog : an introduction to computational linguistics*. Wokingham,England:Addison Wesley Publishing Company.

[Ga+Oa+Jo82] Garnham A., Oakhill, J. and Johnson-Laird P.N. (1982) Referential continuity and the coherence of discourse. *Cognition*,**11**:29-46.

[Gar83] Garnham A. (1983) What's wrong with story grammars? *Cognition* **15**: 145-154.

[Gar87] Garnham A. (1987) *Mental models as representations of discourse and text* (Chichester, Ellis Horward limited).

[Gi+Ho83] Gick M.L. and Holyoak K.J. (1983) Schema induction and analogical transfer. *Cognitive psychology*,**15**:1-38.

[Gol75] Goldman N.M. (1975) Conceptual generation. In R.C. Schank (ed) *Conceptual information processing*. Amsterdam:North Holland (1975) (pagse 289-371)

[Gra78] Grasser A.C. (1978) How to catch a fish: The memory and representation of common procedures.*Discourse processes*,**1**:72-89.

[Gre66] Greimas A.J. (1966) *Semantique structurale*. Paris.

[Gri86] Grishman R. (1986) *Computational linguistics: an introduction*. Cambridge:Cambridge University press

[Gro77] Grosz B.J. (1977) *The representation and use of focus in dialogue understanding*. PhD dissertation, University of California at Berkeley.

[Ha+Be+Sa80] Haberlandt K., Berian C. and Sandson J. (1980) The epsode schema in story processing. *Journal of Verbal learning and Verbal behaviour***19**:635-650.

[Ha+Ha78] Halliday M.A.K. and Hasan R. (1978) *Cohesion in English*. London:Longman.

[Har85] Harris M.D. (1985) *Introduction to natural language processing* Reston,Virginia:Reston Publishing Company, Inc.

[Ha+We73] Hamburger H. and Wexler K.N. (1973) Identifiability of a class of transformational grammar. In K.J.J. Hintikka J.M.E. Moravcsik and P. Suppes (eds) *Approaches to Natural Language* (Dordrecht, Holland: D. Reidel).

[Hob78] Hobbs J.R. (1978) Resolving pronoun references. *Lingua*, **44**:B11-338.

[Joh83] Johnson-Laird P.N. (1983) *Mental models*. Cambridge, MA. Harvard University press.

[Joh91] Johnson-Laird P.N. (1991) Jazz improvisation. In P. Howel, R. West and I. Cross (eds) *Representing music structure*. London:Academic Press.

[Joh93] Johnson-Laird P.N. (1993) *Human and machine thinking*. Hillsdale,NJ:Lawrence Erlbaum Assocaites.

[Jo+Ma80] Johnson N.S. and Mandler J.M. (1980) A tale of two structures: Underlying and surface forms in stories. *Poetics* **9**: 51-86.

[Ki+Gr78] Kintsh W. and Greene E. (1978) The role of culture specific schemata in the comprehension and recall of stories. *Discourse processes*,**1**:1-13.

[Ki+Va75] Kintsch W. and van Dijk T.A. (1975) Recalling and summarizing stories. *Language* **40**: 98-116.

[Ki+Va78] Kintsch W. and van Dijk T.A. (1978) Toward a model of text comprehension and production. *Psychological review*,**85**:363-394.

[Kle73] Klein S., Aeschliman J.F., Balsiger D.F., Converse S.L., Court C., Foster M., Lao R., Oakley J.D., and Smith J. (1973) *Automatic Novel Writing : a status report*. Tech report 186. Madison, Wis: University of Wisconision Computer science dept.

[Lak72] Lakoff G.P. (1972) Structural complexity in fairy tales. *The Study of Man* **1**: 128-190.

[Leb84] Lebowitz M. (1984) Creating characters in a story telling universe. *Poetics*,**13**,171-194.

[Leb85] Lebowitz M. (1985) Story telling as planning and learning. *Poetics*,**14**,483-502.

[Leb87] Lebowitz M. (1987) Planning stories, in *Proceedings of the cognitive science society*, Hillsdale. 234-242.

[Ma+Go82] Mandler J.M. and Goodman M.S. (1982) On the psychological validity of story structure. *Journal of Verbal learning and Verbal behavior*,21:507-523.

[Ma+Jo77] Mandler J.M. and Johnson N.S. (1977) Remembrance of things parsed: story structure and recall. *Cognitive Psychology* **9**: 111-191.

[Ma+Jo80] Mandler J.M. and Johnson N.S. (1980) On throwing out the baby with the bathwater: a reply to Black and Wilensky's evaluation of story grammars. *Cognitive Science* **4**: 305-312.

[Ma+Jo82] Mani K. and Johnson-Laird P.N. (1982) The mental representation of spatial descriptions. *Memory and cognition*,**10**:181-187.

[Man84] Mandler J.M. (1984) *Stories, scripts and scenes:Aspects of schema theory*. Hillsdale,NJ:Lawrence Erlbaum Assocaites, Inc.

[Mar80] Marcus M.P. (1980) *Theory of syntactic recognition for natural language*. Cambridge, Mass:MIT press.

[McK85] McKeown K.R. (1985) *Text generation: using discourse stratagies and focus constraints to generate natural language text.* Cambridge, UK:Cambridge University Press.

[Mee76] Meehan J. (1976) *The metanovel: writing stories by computer.* Ph.d. Dissertation Research report # 76. Computer science department, Yale university.

[Mee81] Meehan J. (1981) Talespin. in [Sc+Ri81], pages 197-227.

[Min75] Minsky, M. (1975) A framework for representing knowledge. In P.H. Winston (ed), *The psychology computer vision.* New haven: Mcgraw-Hill.

[Mo+De85] Mooney M. and De Jong G.F. (1985) Learning schemata for natural language processing. In P.H. Winston (ed), *Proceedings of the ninth international joint conference on artificial intelligence.* Los Angeles, 681-687.

[Ne+Si72] Newell A. and Simon H.A. (1972) *Human Problem Solving* (Englewood Cliffs, NJ: Prentice-Hall).

[Po+Mc+To79] Pollard-Gott L.,McCloskey M. and Todres A.K. (1979) Subjective story structure. *Discourse Processes*,**2**:252-282.

[Pos67] Postal P. (1967) *Constituant structure: A study of contempory models of syntactic description.* Indiana:Indiana University Press.

[Pro68] Propp V. (1968) *Morphology of the Folktale* Austin: University of Texas Press.

[Ri+Sc89] Riesbeck C.K. and Schank R.C. (1989) eds. *Inside case based reasoning* Hillsdale, New Jersey:Lawrence ErlBaum Associates.

[Rum75] Rumelhart D.E. (1975) Notes on a schema for stories. In Bobrow D. G. and A. Collins (eds) *Representation and Understanding : Studies in Cognitive Science* New York: Academic press.

[Rum77] Rumelhart D.E. (1977) Understanding and summarizing brief stories. In D. La Berge and J. Samuels (eds) *Basic Processes in Reading Comprehension* Hillsdale, NJ: Lawrence Erlbaum Associates.

[Rum80] Rumelhart D.E. (1980) On evaluating story grammars. *Cognitive Science,* **4**: 313-316.

[Ru+Or77] Rumelhart D.E. and Ortony A. (1977) The representation of knowledge in memory. In R.C. Anderson, R.J. Spiro and W.E. Montague (eds), *Schooling and the acquisition of knowledge.* Hillsdale, NJ:Lawrence Erlbaum Associates.

[Sac67] Sachs J.S. (1967) Recognition memory for syntactic and semantic aspects of connected discourse. *Perception and Psychophysics*, **2**: 437-442.

[Sc+Ab77] Schank R.C. and Abelson R.P. (1977) *Scripts, Goals, Plans and Understanding.* Hillsdale, NJ:Lawrence Erlbaum Associates.

[Sc+Co73] Schank, R.C. and Colby K. (1973) (eds) *Computer models and thought and language*, San Francisco, C.A.: Wh Freeman and Co.

[Sch72] Schank, R.C. (1972) Conceptual dependency:a theory of natural language understanding. In *Cognitive psychology,***3(4)**:552-631.

[Sch73] Schank, R.C. (1973) Identification and conceptualizations underlying natural language. In [Sc+Co73]

[Sch75] Schank R.C. (1975) The structure of episodes in memory. In D. G. Bobrow and A. Collins (eds) *Representation and Understanding: Studies in Cognitive Science.* New York: Academic Press.

[Sch90] Schank R.C. (1990) *Tell me a story: a new look at real and artificial memory.* New York:Macmillan Publishing Company.

[Sc+Le79] Schank R.C. and Lebowitz M. (1979) *Does a hippie own a hairdryer?*. Tech report 144. Yale University, Department of Computer Science.

[Sc+Ow90] Schank R.C. and Owens C. (1990) The mechanics of creativity. In Kurzwell R. (ed) *The age of intelligent machines.* Mass:MIT press.

[Sc+Ri81] Schank R.C. Riesbeck C.K. (1981) *Inside computer understanding:five programs plus miniatures.* Hillsdale,NJ:Lawrence Erlbaum Associates.

[Sha85] Sharples M. (1985) *Cognition, Computers and Creative writing.* Chichester:Ellis Horwood.

[Shi72] Shirley R. (1972) Poet and program. *Page*, No 25.

[Sid79] Sidner C.L. (1979) Towards a computational theory of definite anaphora comprehesion in English discourse. PhD dissertation, MIT Cambridge, Mass.

[Sid83] Sidner C.L. (1983) Focussing in the comprehesion of definite anaphora. In Brady M and Berwick R.C. *Computational models of discourse.* Cambridge,Mass:MIT press.

[Sim73] Simmons R.F. (1973) Semantic networks: Their computation and use for understanding English sentences. in [Sc+Co73] pages 63-113.

[Ste78] Stenning K. (1978) Anaphora as an approach to pragmatics. In M. Halle, J.W. Bresnan and G.A, Miller (eds), *Linguistic theory and psychological reality.* Cambridge, M.A.:MIT press.

[St+Gl79] Stein N.L. and Glenn C.G. (1979) An analysis of story comprehension in elementary school children. In R. Freedle (ed.) *New Directions in Discourse Processing* Norwood, NJ: Ablex.

[St+Po84] Stein N.L. and Policastro M. (1984) The concept of a story: A comparison between children's and teacher's viewpoints. In H. Mandl, N.L. Stein and T. Trabasso (eds) *Learning and comprehension of text.* Hillsdale,NJ:Lawrence Erlbaum Associates.

[St+Sc+Pa73] Stockwell R.P., Schachter P. and Partree B.H. (1973) *The major syntactic structures of English.* New York:Holt, Rinehart and Winston.

[St+Tr82] Stein N.L. and Trabasso T. (1982) Whats in a story? An approach to comprehension and instruction. In R. Glaser (ed) *Advances in the psychology of instruction* Vol 2. Hillsdale,NJ:Lawrence Erlbaum Associates.

[Tho77] Thorndyke P.W. (1977) Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology* **9**: 77-110.

[Tur63] Turing A. (1963) Computing machinery and intelligence. In E.A. Feigenbaum and J. Feldman (eds) *Computers and Thought*, pages 11-35. New York: Mcgraw-Hill.

[Tur92] Turner S.R. (1992) *Minstrel: A computer model of creativity and storytelling.* PhD thesis presented to the Department of computer science, University of California, Los Angeles,

[Win83] Winograd T. (1983) *Language as a cognitive process vol 1: Syntax.* Reading, MA:Addison-Wesley