

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo 716

April 1983

Understanding Text through Summarization and Analogy

Graziella Tonfoni and Richard J. Doyle

Abstract: Understanding a text exactly in the way that the Text Producer meant the text to be understood is highly unlikely unless the text interpretation process is constrained. Specific understanding-directing criteria are given in the form of a Premise which is a configuration of plot-units. After performing a Premise-directed text summarization, the Text Receiver will have understood the text as the Text Producer intended and will then be able to replace missing relations within the exercises and produce new texts by applying analogy.

This research was done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology.

© Massachusetts Institute of Technology 1983

1. Understanding - Free Interpretation vs. Constrained Interpretation

A text is a communicative linguistic entity which implies a communicative interaction between a Text Producer and a Text Receiver in a specific context. Perfect communication occurs when the set of meanings which the Text Receiver finds in the text are exactly those which the Text Producer wished to convey. Perfect communication is very difficult to achieve. The Text Receiver will most likely interpret the text in a way that is different than what the Text Producer intended. What we propose in this paper is a way to constrain the text interpretation process so that the correct meanings are passed from the Text Producer to the Text Receiver.

A text consists of sequences of information. There is no single way of understanding a text; a text is a highly complex entity which contains many different types of information which need to be selected and classified. In other words, any lexical element or any sentence can be interpreted in more than one way. Text understanding is not based simply on the idea of additive meaning but rather on the idea of meaning organization and selection. Any rather complex text can be understood in several ways depending on the interpretative choices the Text Receiver decides to make. For instance, there is psychological evidence that the Text Receiver (understander) while reading a text will tend to focus his attention on those aspects and those events that affect him or her personally [Car81]. The focus of the understanding process determines which consequences and other events are kept in the representation in memory, and which ones which are deleted or ignored. The selection of information determines how the text will be understood and how it will be summarized. A text will be really understood only when the Text Receiver can understand and summarize it the way the Text Producer intended; this means that the Text Receiver will find and keep only the relevant meanings and relations. Since we are dealing with stories and narrative texts, relevant relations will be expressed always in terms of causal links [Leh81].

We can identify two possible ways of understanding a text.

* Text Understanding means Free Text Interpreting

The Text Receiver has a high level of freedom in understanding the text. In other words, he can select those information sequences which he considers to be focal and ignore those information sequences which he considers to be peripheral. The Text Receiver is allowed to proceed to a free summary production and will not be placed under any kind of constraint when asked to produce an analogous text or to replace missing relations within an exercise [Win82].

* Text Understanding means Constrained Text Interpreting

The Text Receiver will have to respect specific constraints given by the Text Producer in order to understand the text. The Text Receiver will be highly constrained in producing a summarization and in using analogy.

This constrained form of text interpreting is the subject of this paper.

We will use Lehnert's notion of plot-units [Leh81] which are a set of primitive abstractions which capture causal relations among mental states and events. We will propose a set of linking conventions between plot-units, i.e., we will enumerate which plot-units seem to naturally presuppose (links backwards with) other plot-units and which seem to imply (link forwards with) other plot-units. This syntax of plot-units will be the basis of the constraints imposed both on the Text Producer in creating texts, and on the Text Receiver in understanding texts.

To reiterate, there cannot be any reliable transfer of meaning through text unless there are previous restrictions made by the Text Producer (teacher) who is aware of exactly what he wants to point out and have the Text Receiver (student) recognize. The teacher directs the student's attention toward some specific goals so that any summary or application of analogy reflects the same goals. Our idea of text understanding implies goal-oriented text production; goal-oriented text production presupposes specific rules and constraints in the way information is presented (see Section 2). We do not fully agree with Schank's assumption that "the emphasis must be on the content of communication rather than on its form" (see [Sch81]). The content is dependent on the way information is organized. On the one hand there is no *a priori* unique way of giving specific information. On the other hand all possible ways of presenting information are not equivalent because they produce different focuses of attention, different selections of information, different summaries, and finally different ways of understanding. The way the Text Producer wants the meaning to be organized (and the text understood) must be made evident to the Text Receiver.

Thus we introduce the notion of a Premise as a goal-oriented device to be used in understanding a text. The Premise will specify choices made by the Text Producer which must be respected by the Text Receiver. The Premise determines the distinction between focal and peripheral information. The Premise will point out causal-relation configurations at a higher level of abstraction in order to direct summarization and use of analogy.

We will use Lehnert's plot-units within the Premise to direct the understanding and summarization process [Leh81]. A Premise has the following form:

Let Text be a Story about Protagonist's Plot-Unit-Configuration.

The Text Producer directs the attention of the Text Receiver to the following facts:

- * Text T is a Story, meaning a set of Mental States and Events which are causally related.
- * Text T is about one or more Protagonists; Mental States and Events have to be interpreted by adopting the perspectives of one of the Protagonists.
- * Text T is about a specific set of Mental States and Events leading to a specific conclusion or goal. The plot-unit configuration will constrain any summarization or use of analogy.

Any kind of summarization or analogy procedure which is not strictly related to the Premise will be considered inappropriate. Any summarization or use of analogy must be consistent with the goals set out by the Premise; any other possible interpretation which the Text Receiver might produce may be accepted as part of a Free Text-Interpreting Procedure but not as part of a Constrained Text-Interpreting, Text-Understanding Procedure.

To summarize, understanding implies:

The Text Producer must constrain possible interpretations of a text by indicating a specific goal.

The Text Producer must give adequate criteria to interpret the text consistently with the specific goal.

The Text Receiver must be able to identify the goal.

The Text Receiver summarizes the text by respecting the goal-orientation.

The Premise or text-understanding-goal-oriented-device can be more or less specific. As an example, consider a story about two Protagonists interacting in a competitive situation:

Paul competes with Bill to get a job at Atari.
Bill needs but does not have a good resume.
Paul gets the job.
Bill starts a new company.
Paul becomes dissatisfied with his job at Atari.
Paul asks Bill for a job in his new company.
Bill turns Paul down.

This story can have any of the following premises:

Let T be a story about Paul and Bill.
Let T be a story about Paul's and Bill's Efforts.
Let T be a story about Bill's Failure.
Let T be a story about Paul's Success.

Let T be a story about Bill's Problem-Solving Effort.
Let T be a story about Paul's Misleading Success.
Let T be a story about Paul's Problem-Solving Effort.
Let T be a story about Bill's Denial.

These premises proceed from a higher to a lower level of generality and determine different Summaries. Given a text, it is always possible to give a set of possible Premises; Premises divert the focus of attention and result in different selections of information.

2. Well-Formedness Criteria for Text Production

In this section, we outline a set of linguistic constructs which serve to identify plot-unit sequences within a text and isolate them from sequences which are irrelevant. These constructs take the form of connectors between sequences of information. The Text Producer will use these connectors to organize the text in the way that he wants.

Since only those sequences which are part of the plot-unit configuration within the text appear in the Summary, it is possible to produce texts from the same set of information sequences which have highly different meanings, just by altering the use of connectors and the arrangement of the sequences.

The Text Producer must be aware of what he wants the Text Receiver to understand. The Text Producer will have been previously exposed to the notion of plot-units as defined by Lehnert in [Leh81] and must know before producing the text how to classify the information he wants to give in terms of plot-unit configurations. In other words, the Text Producer will know exactly which sequences of information the Text Receiver is to select and retain as part of the Summary and which sequences can be considered as being superfluous and not strictly relevant to the plot-unit configuration within the text.

We will define the latter type of information sequences to be Expansions because they actually expand the text by adding information which is not necessary to understand the text. There is no unique criteria to distinguish strictly relevant information, which is part of the plot-unit configuration, from Expansions, which are not. Any choice of criteria depends exclusively on decisions made by the Text Producer in creating the text. Once made, the decisions must be binding. Only those plot-unit configurations found by using the chosen criteria are valid. The set of criteria is the Premise. Since we are dealing with narrative stories whose

coherence is based on causal relations, the Premise will be expressed as follows:

Let Text be a Story about Protagonist's Plot-Unit-Configuration.

where Story means a set of Mental States and Events which are causally related and which lead to a conclusion. These Events are about one or more Protagonists. The Premise will tell us exactly whose perspective to use, i.e., which Protagonist's viewpoint is relevant. Referring to the example text in Section 1, within a competition situation, Bill's *failure* is equivalent to Paul's *success* and the same sequence can be evaluated in two different ways depending on the Protagonist's point of view and will determine a different plot-unit configuration. Any plot-unit configuration can be simple or complex (see Section 4). To briefly summarize, the choices made by the Text Producer in distinguishing what he thinks is strictly relevant from what is not, are determined by the Text Producer's communicative goals and must be made evident to the Text Receiver.

Correct information-selection is the first necessity towards correct text understanding. Therefore, in producing the text, the Text Producer will respect the following set of rules:

- * The text must be a Story; this means a narration of a causally ordered chain of Events and Mental States which are about one or more Protagonists.
- * The text can be short or long, and of moderate complexity, but the Text Producer must use only five connecting elements in order to link the sequences. These five elements are:
 - * *null (start of sequence)* -- used to recognize sequences which are relevant to text understanding because they are part of the plot-unit configuration; either simple plot-units or the initial component of a complex plot-unit.
 - * *so* -- used to recognize sequences which are part of a complex plot-unit and to express continuity and causal ordering.
 - * *but* -- used to recognize sequences which are part of a complex plot-unit and to represent causal contraposition.
 - * *and* -- used to separate a plot-unit sequence from its expansion or expansions; defines the end of the plot-unit sequence. Any *and*-sequence has to be deleted during the summarization procedure. *And*-sequences represent conjunctions.
 - * *because* -- used to separate a plot-unit sequence from its expansion or expansions; defines the end of the plot-unit sequence. Any *because*-sequence also has to be deleted during the summarization procedure. *Because*-sequences represent indirect causal relations.

- * the *so* connector represents same-goal orientation within a plot-unit configuration.
- * the *but* connector represents different goal orientation within a plot-unit configuration.
- * the text can allow a high level of redundancy in information; any kind of information must however, be previously distinguished and subdivided into relevant information within plot-unit sequences and irrelevant information within expansions.
- * the subject (Protagonist) of any sequence has to be explicitly given. We will use the convention that the subject is always the first element in a sequence.
- * the Text Producer will repeat the subject rather than use coreferential pronouns.
- * any reflexive or possessive constructs must also contain an explicit subject (again, no pronouns).
- * the Text Producer must provide a list of verbs associated with each plot-unit he wishes the Text Receiver to recognize. This is an extremely simple form of semantics.
- * the verb is always the second element of a sequence (immediately following the subject).

By respecting the above set of conventions, the Text Producer ensures that the Text Receiver will be able to produce a correct Summary by recognizing connectors and identifying the plot-unit configuration within the text.

The first example in the Appendix shows the separation of Expansions through the use of connectors and the recognition of simple plot-units.

The second example shows how different connectors and different distributions of the same information sequences can produce different Summaries.

3. TUP. Text-Understanding Procedures

Up to this point, we have not stated whether the Text Producer and Text Receiver are intended to be persons or AI programs. As of now, we have not implemented a program to play the role of Text Producer but we believe that this is quite possible and have relegated this task to future work. However, we have implemented a program to play the role of Text Receiver. The Appendix presents the program itself and some examples.

In this section, we outline a third entity: TUP or Text-Understanding Procedures which interfaces between the Text Producer and the Text Receiver. The TUP program can understand a text and produce

Summaries in the way that the Text Receiver is intended to but in addition, TUP can generate exercises from the text for the Text Receiver to solve through the use of analogy. As of now, the text understanding part of TUP has been implemented, while the exercise-generating part has not.

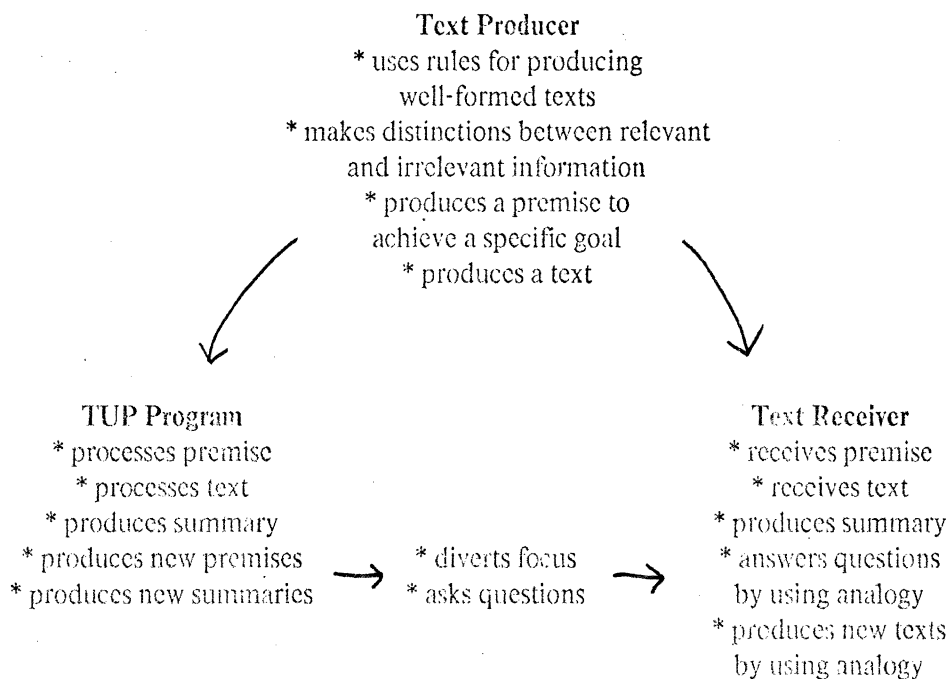
Our Text-Understanding Procedures are based on the following linguistic assumptions:

- * a text (story) does not have only a single meaning but can be interpreted in different ways depending on the number of Protagonists which are involved, the perspective we are going to assume, and the conclusion we want to reach.
- * in order to understand a text, the Text Receiver must summarize it; this means he must be able to recognize the relevant information and remove the irrelevant.
- * understanding is a multi-level process which involves the following steps:
 - * rules for text well-formedness.
 - * guiding of understanding through use of a Premise.
 - * analogy application in question-answering and missing relations replacement.

Consider the scheme in Fig. 1. This scheme embodies the following procedures:

- * The Text Producer produces a text which is well-formed with respect to the given set of rules and the

Fig. 1. Text-Understanding Procedures



stated goals.

- * The Text Producer will use the plot-units as defined in [Leh81] in order to produce the text and will provide the set of plot-units as well as the text.
- * The plot-units can be recognized by the use of a list of verbs, one of which has to be within a sequence which corresponds to a given plot-unit.
- * All plot-unit sequences must have an associated Protagonist; all the Protagonists which appear in a text are given as part of the Premise.
- * Links between plot-units will be previously given also.
- * The text will be evaluated by TUP in two different steps:
 - * TUP finds expansions through recognition of the appropriate connectors and removes them.
 - * TUP isolates plot-unit sequences by subject and verb recognition and produces the summary.
- * Through summarization, TUP can produce all possible Premises for the text.
- * By referring back to the Premises and using plot-unit configurations and links, TUP produces the correct Summaries (those intended by the Text Producer).
- * TUP will be able to use different Summaries together with their corresponding Premises to create exercises (reduced versions of the same text) where there will be one missing relation which can be inserted by using analogy and referring to precedents.
- * The Text Receiver, after having applied analogy to insert or replace relations will be able to recognize plot-units and links between them which are consistent with to different Premises. He will be able to create new texts which can have different elements but keep the same relations.

To conclude, the aim of the TUP program is to:

- * Achieve text understanding by correct summarization through the use of plot-unit configurations.
- * Generate exercises in the spirit of exercises and precedents [Win81] to be solved by analogy. This could involve insertion of missing relations within a complex plot-unit or new text production by substituting elements while keeping the same plot-unit configuration.

4. Syntax for Plot-Units

As already stated, this work refers to Lehnert's plot-units [Leh81] with the following two changes:

- * Positive and negative Mental States and Events have been substituted by the notion of same-goal orientation (same sign) and different-goal orientation (opposite sign).
- * We have considered the syntactic aspects of plot-units and plot-unit aggregations (simple plot-units, complex plot-units, and plot-unit aggregations) by identifying different combinatorial possibilities (or ways of linking plot-units) based on psychological evidence [Ton82e].

Plot-unit recognition and text summarization are necessary prerequisites to understanding a text. Only after a summarization procedure [Leh81] has been performed, is it possible to proceed to Relevant Question Formation, Correct Question Answering, and Analogy Procedure Application (see Sections 5 & 6).

The text produced by the Text Producer exhibits a configuration of plot-units which can be simple, complex, or aggregated. We define aggregation of plot-units as a set of two or more complex plot-units which are causally related or nested.

The simple plot-units which we use are shown in Fig 2.

The complex plot-units are shown in Fig. 3.

An example of a plot-unit aggregation is:

Protagonist A's *intentional problem resolution* of Protagonist B's *giving up*.

Fig. 2. Simple Plot-Units

<i>motivation</i>	<i>success</i>	<i>failure</i>	<i>change of mind</i>
<i>loss</i>	<i>mixed blessing</i>	<i>perseverance</i>	<i>resolution</i>
<i>hidden blessing</i>	<i>enablement</i>	<i>negative tradeoff</i>	<i>positive tradeoff</i>
<i>problem</i>	<i>complex positive event</i>	<i>complex negative event</i>	

Fig. 3. Complex Plot-Units

intentional problem resolution
success born of adversity
starting over
sacrifice
killing two birds

fortuitous problem resolution
fleeting success
giving up
nested subgoals

The first concern in recognizing simple plot-units, complex plot-units, or plot-unit aggregations is establishing some syntactic constraints which are in some cases based on psychological evidence [Ton82e]. In other words, there is evidence that there are plot-unit combinations which are more natural and recognizable and others which are less plausible.

We propose a set of conventions for linking plot-units, i.e., we enumerate which plot-units seem to presuppose (links backwards with) other plot-units and which plot-units imply (link forwards with) other plot-units. It is this syntax of plot-units that the Text Producer must adhere to in constructing goal-oriented texts for the Text Receiver to understand. By using the syntax, the Text Producer can take advantage of predispositions in the mind of the Text Receiver on how plot-units (or at a more primitive level, mental states and events) can be causally related. The syntax of plot-units is meant to reflect these predispositions.

For simple plot-units only, we identify in Fig. 4 and Table I the set of linking relations which combine plot-units by respectively, backward, or forward chaining.

For example, *success* can presuppose *effort* and *motivation* can imply *perseverance*.

This set of links is a subset of the set of all possible links between plot-units. Those links which do not appear in this set are considered implausible and should not appear in a text.

The choice of the words "presuppose" and "imply" to represent respectively, backward linking, and forward linking is not accidental, for there is a qualitative difference between the two types of links. Backward links seem to be stronger in the sense that the temporally earlier plot-unit is very likely to be present when the temporally later plot-unit is observed. With forward links, the later plot-unit may be present, but there is nothing particularly compulsory about the link.

Furthermore, there is a subset of the set of backward links which are particularly strong, and whose existence has been shown by evidence from psychological experiments [Ton82e].

We classify and define them in Fig. 5.

Whenever the second plot-unit in one of these links appears in the text, the Text Receiver may

Fig. 4. Backward-Chaining Links for Simple Plot-Units

effort <== success	effort <== loss
motivation <== success	failure <== problem
motivation ==> enablement	enablement <== change-of-mind
failure <== motivation	problem <== success
enablement <== success	success <== misleading-success
enablement <== resolution	problem <== effort
problem <== loss	problem <== change-of-mind
problem <== resolution	problem <== perseverance
change-of-mind <== problem	success <== change-of-mind
problem <== loss	success <== mixed-blessing
perseverance <== motivation	complex-negative-event <== loss
problem <== resolution	failure <== hidden-blessing
loss <== hidden-blessing	complex-negative-event <== hidden-blessing
loss <== negative-tradeoff	failure <== negative-tradeoff
positive-tradeoff <== success	positive-tradeoff <== resolution
complex-negative-event <== failure	success <== complex-positive-event
complex-positive-event <== mixed-blessing	complex-positive-event <== resolution
complex-positive-event <== positive-tradeoff	complex-negative-event <== negative-tradeoff

Table I. Forward-Chaining Links for Simple Plot-Units

	m	s	f	com	l	mb	pv	r	hb	e	nt	cpe	pr	pt	cne
motivation		*			*		*			*					
success				*		*	*	*				*		*	
failure	*				*		*		*		*		*		*
change of mind					*				*		*		*		*
loss				*			*	*	*		*		*		*
mixed blessing							*				*		*		
perseverance		*			*										
resolution		*			*	*									
hidden blessing		*			*										
enablement		*						*				*			
negative tradeoff					*				*						
complex positive event		*				*		*						*	
problem		*		*	*		*	*							*
positive tradeoff		*				*						*			
complex negative event					*	*			*				*		

Fig. 5. Trigger Plot-units

failure <== problem	problem <== success
problem <== resolution	success <== misleading success
problem <== effort	effort <== success
enablement <== success	problem <== enablement
problem <== change of mind	problem <== loss
success <== mixed blessing	failure <== hidden blessing
loss <== hidden blessing	

immediately search backward for the first. We define the second plot-unit as a Trigger, because these plot-units immediately trigger a search for the associated plot-unit by backward-linking.

There may also be other information sequences corresponding to plot-units, which appear in the text between the two plot units in one of these links. The presence of such intervening plot-units does not deny the existence of the link; the two plot-units in a link must be only ordered, not necessarily consecutive. This intervening "noise" can be ignored in order to produce the correct links.

Considering Lehnert's complex plot-units [Leh81], these form another category of linking relations which are more concrete than the primitive links; they are in fact constructed from the primitive links. These complex plot-units involve both forward and backward links and are shown in Fig. 6.

Trigger plot-units identify strong combinatorial possibilities through backward-chaining; combinatorial possibilities through forward-chaining are weaker. In general, the thirteen Trigger backward-chaining links are unique, while the set of forward-chaining involve a set of possible links rather than a unique one. By

Fig. 6. Complex Plot-Units

way-out	failure <== problem <== effort
intentional problem resolution	problem <== success <== resolution
fortuitous problem resolution	problem <== resolution
success born of adversity	problem <== success
fleeting success	success <== loss
starting over	success ==> loss ==> problem <== perseverance
giving up	failure <== problem <== change of mind
sacrifice	success ==> negative tradeoff
nested subgoals	motivation <== success <== success
killing two birds	complex positive event <== success <== success

scanning Table I, it is possible to notice the differing combinatorial power of the simple plot-units with regard to forward chaining. In other words, each simple plot-unit triggers some different set of expectations (predictions) about which other plot-units may possibly follow. This linking is much less deterministic than the backward linking of the Trigger plot-units.

To summarize, we have enumerated the ways in which plot-units may be causally linked. We have seen that backward-chaining is a stronger syntactic configuration than forward chaining.

How the syntax for plot-units is organized in a database for the program outlined in Section 3 is shown in the Appendix.

5. Relevant Question Formation - Correct Question Answering

In order to obtain correct answers from the Text Receiver, the Text Producer along with the TUP program must ask Relevant Questions. A Relevant Question typically takes the form of a question about a relation which is missing from the text but which is expected to be there because it is implied by the plot-unit syntax defined in Section 4. A Relevant Question usually refers to an exercise which has been generated by the TUP program. Relevant Question Formation is directly related to the rules for Well-Formedness (see Section 2) for a text.

In other words, any question about a text must refer back to:

- * a specific goal towards which the text has been oriented (given by the plot-unit configuration in the Premise).
- * the Summary of the text (also determined by the Premise).
- * the syntax of Trigger plot-units (backward links only).
- * the syntax of complex plot-units (both backward and forward links).

Again, Relevant Questions concerning missing relations identification must respect the following set of conditions:

- * questions must be Premise-related.

- * questions must be Summary-related.
- * questions may ask for insertions which are implied by backward links of Trigger plot-units.
- * questions may ask for insertions which are implied by backward or forward links involving a simple plot-unit within a complex plot-unit configuration.
- * when asking for insertions which are forward-directed, the Text Producer has to be aware that these are only plausible links, as opposed to psychologically justified ones [Ton82c]. (As an example, there is psychological evidence that a *problem* naturally links backwards with a *failure*, while it is merely possible that given a *problem* there may be a forward link to *success* or *loss*).
- * questions should not imply inferences which refer to expansions, only to the summary (the plot-unit configuration within the text).

Consider the following example:

Let T be a story about Paul's *giving-up*.

Paul wanted to send flowers to Mary, because Mary had graduated in Linguistics from M.I.T.
Paul didn't have Mary's address because Paul didn't have his address book and Paul only knew that Mary was living on campus.
Paul gave up because Paul didn't want to call the Linguistics Dept. to find out Mary's address.

Summary:

Paul wanted to send flowers to Mary.
Paul didn't remember Mary's address.
Paul gave up.

Consider the questions in Fig. 7 which ask for the insertion of missing relations.

Misleading questions from the Text Producer initiate incorrect inferences by the Text Receiver, leading to wrong answers. Relevant Question Formation and Correct Question Answering must precede any Analogy Procedure Application (Section 6). Since we are dealing with relational configurations (plot-units) and more specifically, Premise-determined, relational-configuration-based Summaries, Relevant Question Formation will aim at relation insertion rather than element insertion. Relevant Question Formation must precede any exercise production by the TUP program. We are assuming an exercise (as in [Win81]) to be a reduced version of the text with some relations removed.

We modify Winston's exercises in the following aspects:

- * the exercise has to be derived from the summary and not directly from the text (has to be Premise- or

Fig. 7. A Correctly and an Incorrectly Formed Question

Question 1

Paul wanted to send
flowers to Mary

?
insert what happened

Paul gave up

Question 2

?
insert what happened

Paul only knew that
Mary was living on campus

Paul gave up

Question 1 is well-formed because:

- * Summary is used.
- * syntax of complex plot-units is used
giving up = (failure <== problem <== change of mind)

Question 2 is ill-formed because:

- * an Expansion is used.

plot-unit-configuration-based).

- * the missing relations (simple plot-units) must be backwards linked with a Trigger plot-unit or must be components of complex plot-units whose other components do appear in the Summary.

Since we assume that these syntactic configurations are known, the Text Receiver will be able to avoid confusion and insert missing relations in the following way:

- * recognize plot-unit configurations in the Premise by looking at the Summary.
- * search for backward-chaining links of Trigger plot-units or for links within complex plot-unit configurations.
- * insert the missing simple plot-unit or relation.

In other words, these conditions are well-formedness conditions for Exercise Production and Relevant Question Formation and must be satisfied before proceeding to Correct Question Answering and Analogy Procedure Application.

6. Analogy Procedure Application

Any analogy procedure which is applied to text understanding has to be based on a previous summarization procedure on another text. The use of analogy in text understanding can be directed toward any of the three following goals:

- (1) Recognition of analogous texts.
- (2) Insertion of missing relations within an Exercise by referring back to the original text.
- (3) Production of new texts which keep the same relational structure of the original text, but change the elements.

We will now consider these three uses of analogy in more detail.

- (1) Analogy can be applied at different levels of abstraction. We identify three levels of analogy for texts:

- (a) two narrative texts are strongly analogous when they have exactly the same plot-unit configuration and the same expansions.
- (b) two narrative texts are weakly analogous when they have exactly the same plot-unit configuration but not necessarily the same expansions.
- (c) two narrative texts are misleadingly analogous and only similar when they have the same expansions but a different plot-unit configuration.

An example of three different levels of analogy is given by the texts CONDO1, CONDO2, and CONDO3 (see the third example in the Appendix) which are graphically represented in Fig. 8.

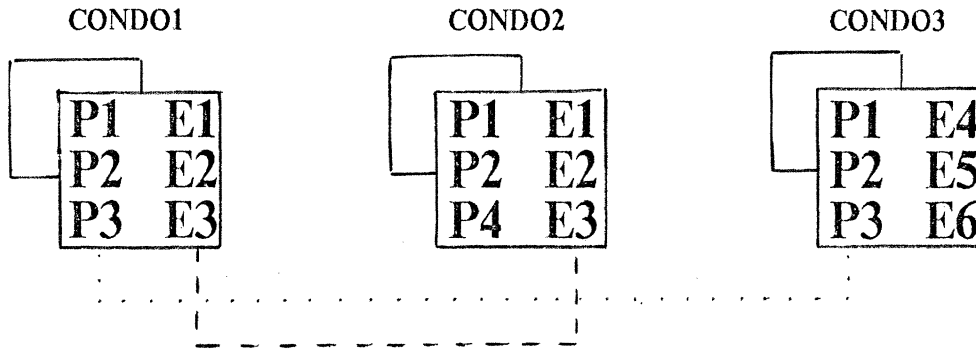
The three texts can be given the following premises, respectively:

Let T be a Story about Paul's *giving up* (failure, problem, change of mind).
Let T be a Story about Paul's *intentional problem resolution* (failure, problem, resolution).
Let T be a Story about Paul's *giving up* (failure, problem, change of mind).

CONDO1 and CONDO2 are misleadingly analogous because their expansions match but not their plot-units. CONDO1 and CONDO3 are weakly analogous because their plot-units match but not their expansions. CONDO2 and CONDO3 do not match under any of the specified criteria.

- (2) Missing relations insertion within an Exercise has been discussed already in Chapter 5. The use of

Fig. 8. Three Forms of Analogy



P's are plot-units.
E's are expansions.

strongly analogous —————
weakly analogous
similar -----

analogy is relevant to this process as well. To identify missing relations, the entire plot-unit configuration made explicit in the Summary has to be matched against the established configurations of complex plot-units. We may consider this process as a form of global analogy. When we insert missing relations into expansions rather than into the plot-unit configuration (Summary) we must restrict matching to single plot-units and their expansions. This is a more localized form of analogy.

Consider again the text CONDO3 and the following Exercise:

Paul wanted to buy a condominium because Paul was sick of Paul's apartment.

We isolate one plot-unit and its expansion and ask for replacement of the expansion. When plot-units are replaced or inserted there may be available strong backward-linking relations to constrain the choice of which plot-unit may be replaced or inserted. But when expansions are replaced, we can at best only guess which links may be valid. The operations on expansions are necessarily local; we are not allowed to use the links which are defined only for plot-units.

In other words, local analogy (involving expansions) is weak, and two texts should be considered properly analogous only when their plot-units match.

(3) This final section deals with the production of analogous texts. Since analogy can apply at different

levels, we will have to specify each time to which kind of analogy we are referring.

Analogous text production means:

- * keeping the same Premise,
- * keeping the same plot-unit configuration which is in the precedent text,
- * adding or replacing expansions.

Consider the following three texts:

(a) Let T be a Story about Paul's *giving up*.

Paul wants to buy flowers for Mary, because today is Mary's birthday.
Paul doesn't have enough money because Paul lost money playing cards.
Paul gives up because Paul doesn't want to borrow any money.

(b) Let T be a Story about Paul's *fortuitous problem resolution*.

Paul wants to buy flowers for Mary, because today is Mary's birthday.
Paul doesn't have enough money, because Paul lost money playing cards.
Paul finds some flowers and brings them to Mary,
because Paul doesn't want to borrow any money.

(c) Let T be a Story about Jim's *giving up*.

Jim wants to fly to I.A. because there is a conference Jim is interested in.
Jim forgets to make a reservation to get a cheap flight because Jim is lazy.
Jim decides Jim cannot afford the full fare because Jim doesn't have enough money.

Texts (a) and (b), which are apparently analogous are actually only similar (as we have defined similar) because they do not produce the same Summary (plot-unit configuration) and they have different Premises.

On the other hand, texts (a) and (c) which are about different situations and Protagonists, are analogous, because they have the same Premise and the same plot-unit configuration in their Summaries.

To conclude, Analogous Text Production requires plot-unit recognition and correct summarization of the precedent text to preserve the causal chain of relations within the text as a template for generating new and analogous texts.

7. ACKNOWLEDGEMENTS

For valuable discussions about the ideas in this paper, we would like to thank Marvin Minsky, Wendy Lehnert, Dedre Gentner, Hayward Alker, and Daniel Schneider.

8. REFERENCES

- [BlW79] Black, J.B. and Wilensky, R., "An Evaluation of Story Grammars", *Cognitive Science* 3, no. 3, 213-230, 1979.
- [Car79] Carbonell, J. G., *Inside Computer Understanding*, Lawrence-Erbaum Associates Publishers, Hillsdale, New Jersey, 1979.
- [Car82] Carbonell, J. G., "Metalanguage Utterances in Purposive Discourse", Carnegie-Mellon C.S. Memo 82-125, 1982.
- [DyL80] Dyer, M., Lehnert, W., "Memory Organization and Search Processes", Dept. of Comp. Sci. Research Report 175, Yale Univ., 1980.
- [Gen82] Gentner, D., "The Structure of Analogical Models in Science", BBN Report 4451, Cambridge, 1982.
- [Gen80] Gentner, D., "A Structure-Mapping Approach to Analogy and Metaphor", *Proceedings of the International Conference on Cybernetics and Society*, 75-79, 1980.
- [Leh81] Lehnert, W. G., "Plot-Units and Narrative Summarization", *Cognitive Science* 4, 293-331, 1981.
- [LAS83] Lehnert, W. G., Alker, H. R., Schneider, D. K., "The Heroic Jesus: The Affective Plot Structure of Toynebee's *Christus Patiens*", *draft*, MIT Political Science Dept.
- [LDJ80] Lehnert, W. G., Dyer, M. G., Johnson, P. N., Young, C. J., Harley, S., "An In-Depth Understanding of Narratives", Yale Dept. of Computer Science Technical Report 188, December 1980, also *Artificial Intelligence*, to appear.
- [Min75] Minsky, M., "A Framework for Representing Knowledge", in Winston, P. H. (ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, 211-277, 1975.
- [Min80] Minsky, M., "K-Lines: A Theory of Memory", *Cognitive Science* 4, 117-133, 1980.
- [Min82a] Minsky, M., "Why People Think Computers Can't", *AI Magazine*, 3-15, 1982.
- [Min82b] Minsky, M., "On Meaning", *draft*, MIT AI Laboratory.
- [Sch81] Schank, R. C., *Inside Computer Understanding*, Lawrence-Erbaum Associates, Hillsdale, New Jersey, 1981.
- [Ton82a] Tonfoni, G., "Conditions and Constraints on Analogy Procedure Applications", *Lingua e Stile*, to appear, Il Mulino, Bologna, 1982.
- [Ton82b] Tonfoni, G., "Paradoxes and Censors", *Semiotica*, to appear, Indiana Univ. Press.
- [Ton82c] Tonfoni, G., *A Frame-System Theory of Text*, *draft*, MIT AI Laboratory.

[Ton82d] Tonfoni, G., *Dalla Linguistica del Testo alla Teoria Testuale*, Unicopli, Milano, 1982.

[Ton82e] Tonfoni, G., *Per una Ipotesi Didattica Testuale*, BQ, Modena, 1982.

[Ton83] Tonfoni, G., "Syntax for Plot-Units", BBN Report, *to appear*.

[Win80] Winston, P. H., "Learning and Reasoning by Analogy", *CACM*, 23, no. 12, 689-703, 1980.

[Win81] Winston, P. H., "Learning New Principles from Precedents and Examples", MIT AI Laboratory Memo 632, May 1981.

[WiH81] Winston, P. H., and Horn, B. K. P., *LISP*, Addison-Wesley, 1981.

Appendix I - The Program and some Examples

```
:: TEXT-UNDERSTANDING PROCEDURES
```

```
(defvar *protagonists*)
```

```
(defun summarize (text)
```

```
;; produces a summary (plot-unit configuration) of the text  
;; each sequence is summarized as a protagonist and a simple plot-unit  
(find-plot-units (strip-expansions text)))
```

```
(defun strip-expansions (text)
```

```
;; removes expansions from the text, leaving the sequences which are  
;; part of the plot-unit configuration  
(loop for sentence in text  
  unless (or (equal sentence 0)  
             (memq (car sentence) '(because and)))  
  collect (loop for word in sentence  
    until (memq word '(because and))  
    collect word)))
```

```
(defun find-plot-units (text)
```

```
;; identifies the protagonist and the simple plot-unit for each sequence  
;; in the text after expansions have been removed  
(loop for sentence in text  
  with subject and verb  
  collect (if (memq (car sentence) '(but so))  
             (setq sentence (cdr sentence)))  
            (setq subject (car sentence))  
            (setq verb (cadr sentence))  
            (cons subject  
                  (ncons (loop for plot-unit in *plot-units*  
                            do (if (memq verb (get plot-unit 'verbs))  
                                (return plot-unit)))))))
```

```
(defun find-expansions (text)
```

```
;; isolates the expansions in the text  
(loop for sentence in text  
  unless (or (equal sentence 0)  
            (and (not (memq 'because sentence))  
                 (not (memq 'and sentence))))  
  collect (loop for words on sentence  
    when (memq (car words) '(because and))  
    do (return (cdr words))))
```

```
(defun find-links (summary)
```

```
;; consults a list of all valid links between simple plot-units and
```

```
;; identifies those which appear in the text
(loop for protagonist in *protagonists*
  append (loop for link in *links*
    with pos1 and pos2
    when (and (setq pos1
      (position (list protagonist
        (car link)
        summary))
      (setq pos2
        (position (list protagonist
          (caddr link)
          summary))
        (< pos1 pos2))
      collect (cons protagonist link))))))

(defun position (element list)
  ;; finds the position of element in list
  (if (member element list)
    (- (1+ (length list)) (length (member element list)))))

(defun find-complex-plot-units (links)
  ;; consults a list of complex plot-units and identifies those which
  ;; appear in the text by verifying that all their component links
  ;; are present
  (loop for protagonist in *protagonists*
    append (loop for complex-plot-unit in *complex-plot-units*
      when (loop for link
        on (get complex-plot-unit 'form)
        by 'caddr
        until (null (cdr link))
        always (member (list protagonist
          (car link)
          (cadr link)
          (caddr link)
          links))
        collect (list protagonist complex-plot-unit))))))

(defun strongly-analogous? (text1 text2)
  ;; two texts are strongly analogous if their summaries (plot-unit
  ;; configurations) and their expansions match
  (if (and (equal (summarize text1) (summarize text2))
    (equal (find-expansions text1) (find-expansions text2)))
    t))

(defun weakly-analogous? (text1 text2)
  ;; two texts are weakly analogous if their summaries match
  ;; (but not necessarily their expansions)
  (if (equal (summarize text1) (summarize text2)) t))
```



```
(defun similar? (text1 text2)
  ;; two texts are similar if their expansions match
  ;; (but not necessarily their summaries)
  (if (equal (find-expansions text1) (find-expansions text2)) t))
```

```
:: SYNTAX FOR PLOT-UNITS
```

```
(defvar *plot-units*)
```

```
(defvar *links*)
```

```
(defvar *complex-plot-units*)
```

```
(defvar *triggers*)
```

```
(setq *plot-units*
```

```
;; list of simple plot-units
```

```
'(motivation failure enablement problem success change-of-mind  
  loss mixed-blessing perseverance resolution  
  hidden-blessing negative-tradeoff  
  complex-positive-event positive-tradeoff  
  effort misleading-success complex-negative-event))
```

```
(setq *links*
```

```
;; valid links between simple plot-units (forward and backward)
```

```
'((effort < success) (effort < loss) (motivation < success)  
(motivation > perseverance) (motivation > enablement)  
(failure < problem) (failure < motivation) (failure > hidden-blessing)  
(enablement < success) (enablement < change-of-mind)  
(enablement < resolution) (problem < success) (problem < loss)  
(problem > perseverance) (problem < resolution)  
(success < misleading-success)  
(problem < resolution) (problem < effort)  
(change-of-mind < problem) (success < change-of-mind)  
(problem < loss)  
(success < mixed-blessing)  
(complex-positive-event < mixed-blessing)  
(perseverance < motivation) (problem < perseverance)  
(problem < resolution)  
(failure < hidden-blessing) (loss < hidden-blessing)  
(complex-negative-event < hidden-blessing) (loss < negative-tradeoff)  
(failure < negative-tradeoff) (success < complex-positive-event)  
(complex-positive-event < resolution) (problem < change-of-mind)  
(complex-positive-event < positive-tradeoff)  
(positive-tradeoff < success) (positive-tradeoff < resolution)  
(complex-negative-event < failure) (complex-negative-event < loss)  
(complex-negative-event < negative-tradeoff)))
```

```
(setq *complex-plot-units*
```

```
;; list of complex plot-units
```

```
'(intentional-problem-resolution fortuitous-problem-resolution  
  success-born-of-adversity fleeting-success starting-over  
  way-out giving-up sacrifice nested-subgoals killing-two-birds))
```

```
;; structure (component links) of complex plot-units
(defprop way-out
  (failure < problem < effort)
  form)
(defprop intentional-problem-resolution
  (problem < success < resolution)
  form)
(defprop fortuitous-problem-resolution
  (problem < resolution)
  form)
(defprop success-born-of-adversity
  (problem < success)
  form)
(defprop fleeting-success
  (success > loss)
  form)
(defprop starting-over
  (success > loss > problem < perseverance)
  form)
(defprop giving-up
  (failure < problem < change-of-mind)
  form)
(defprop sacrifice
  (success > negative-tradeoff)
  form)
(defprop nested-subgoals
  (motivation < success < success)
  form)
(defprop killing-two-birds
  (complex-positive-event < success < success)
  form)

(setq *triggers*
  ;; list of trigger plot-units
  '((failure < problem) (motivation < failure) (failure < hidden-blessing)
    (enablement < success) (enablement < resolution) (problem < success)
    (problem < loss) (success < mixed-blessing)
    (motivation < perseverance) (problem < resolution)
    (success < positive-tradeoff) (resolution < positive-tradeoff)))
```

```
;; EXAMPLE 1: the ATARI text
```

```
;; This example shows the separation of expansions through the use of  
;; connectors and the recognition of simple plot-units
```

```
; read text  
(load 'atari)
```

```
(setq text  
  (Paul competes with Bill to get a job at ATARI and  
    Paul thinks Paul might win because Paul presented a really impressive  
    curriculum)  
  (Bill needs-but-does-not-have a good curriculum because  
    Bill was too young and didnt have much experience)  
  (And you cannot  
    have much experience if you dontt work at least for two years in a big  
    company)  
  (Paul gets the promotion so Bill decides to leave ATARI to  
    start a new company which Bill calls Computer World)  
  (Bill succeeds with his  
    company because Bill is very good in public relations  
    and in organization)  
  (Paul becomes-dissatisfied with Paul's job at  
    ATARI because Paul wants to have another promotion but couldnt have  
    the promotion so Paul goes-and-asks Bill for a job)  
  (Bill turns-down Paul because Bill remembered about ATARI  
    and wanted to have revenge)))
```

```
; read the list of verbs which are associated with and identify  
; simple plot-units  
(load 'semanticsc)
```

```
(defprop effort (competes takes-part-in-a-race tries-to-get) verbs)  
(defprop problem (needs-but-does-not-have has-to-do-but-is-not-able  
  tries-to-achieve-but-does-not-know-how) verbs)  
(defprop success (gets succeeds wins obtains) verbs)  
(defprop problem-solving-effort (decides-to-give-up  
  changes-his-mind goes-and-asks) verbs)  
(defprop misleading-success (becomes-unhappy becomes-dissatisfied) verbs)  
(defprop perseverance (turns-down denies) verbs)
```

```
(setq *protagonists* '(paul bill))
```

```
(strip-expansions text)  
((PAUL COMPETES WITH BILL TO GET A JOB AT ATARI)  
(BILL NEEDS-BUT-DOES-NOT-HAVE A GOOD CURRICULUM)  
(PAUL GETS  
  THE  
  PROMOTION
```

SO
BILL
DECIDES
TO
LEAVE
ATARI
TO
START
A
NEW
COMPANY
WHICH
BILL
CALLS
COMPUTER
WORLD)

(BILL SUCCEEDS WITH HIS COMPANY)

(PAUL BECOMES-DISSATISFIED WITH PAUL+S JOB AT ATARI)

(BILL TURNS-DOWN PAUL))

(find-expansions text)

((PAUL THINKS

PAUL
MIGHT
WIN
BECAUSE
PAUL
PRESENTED
A
REALLY
IMPRESSIVE
CURRICULUM)

(BILL WAS TOO YOUNG AND DIDN+T HAVE MUCH EXPERIENCE)

(YOU CANNOT

HAVE
MUCH
EXPERIENCE
IF
YOU
DON+T
WORK
AT
LEAST
FOR
TWO
YEARS
IN
A
BIG

COMPANY)
(BILL IS VERY GOOD IN PUBLIC RELATIONS AND IN ORGANIZATION)
(PAUL WANTS
TO
HAVE
ANOTHER
PROMOTION
BUT
COULDN'T
HAVE
THE
PROMOTION
SO
PAUL
GOES-AND-ASKS
BILL
FOR
A
JOB)
(BILL REMEMBERED ABOUT ATARI AND WANTED TO HAVE REVENGE))

; each sequence is summarized as a protagonist and a simple plot-unit
(setq summary (summarize text))

((PAUL EFFORT)
(BILL PROBLEM)
(PAUL SUCCESS)
(BILL SUCCESS)
(PAUL MISLEADING-SUCCESS)
(BILL PERSEVERENCE))

; identify which links between simple plot-units are present
(setq links (find-links summary))

((PAUL EFFORT < SUCCESS)
(PAUL SUCCESS < MISLEADING-SUCCESS)
(BILL PROBLEM < SUCCESS)
(BILL PROBLEM > PERSEVERENCE)
(BILL PROBLEM < PERSEVERENCE))

:: EXAMPLE 2: The VACATION text

:: This example shows how different connectors and different distributions
:: of the same information sequences can produce different summaries

; read the texts
(load 'vacation)

```
(setq deep
 '((John decided to have a vacation)
  (John was very tired)
  (John hadn't had any break for months)
  (John went to a travel agency to buy a ticket)
  (John forgot John's wallet)
  (John didn't used to care about money)
  (John gave up)
  (John was too lazy to go back and get the money for the ticket)))
```

```
(setq deep1
 '(0
  (John decided-to-have a vacation
   because John was very tired
   because John hadn't had any break for months)
  0
  (John went-to-buy a ticket at a travel agency)
  (But John forgot John's wallet
   because John didn't used to care about money)
  0
  (John gave-up
   because John was too lazy to go back
   and get the money for the ticket)))
```

```
(setq deep2
 '(0
  (John was-tired)
  (So John decided-to-have a vacation
   because John hadn't had any break for months)
  0
  (John went-to-buy a ticket at a travel agency
   and John didn't used to care about money)
  (John forgot John's wallet)
  0
  (John was-too-lazy to go back
   and get the money for the ticket)
  (So John gave-up)))
```

```
(setq deep3
 '(0
  (John hadn't-had any break for months)
```

```
(So John was-tired)
(So John decided-to-have a vacation)
0
(John didn't-used-to-care about money)
(So John forgot John's wallet)
0
(John went-to-buy a ticket at a travel agency)
0
(John was-too-lazy to go back and get the money for the ticket)
(So John gave-up)))

; read the list of verbs which are associated with and identify simple
; plot-units for the first text
(load 'semanticsd1)

(setq *protagonists* '(john))

(defprop motivation (decided-to-have wanted-to-have intended-to) verbs)
(defprop effort (went-to-buy left-to-get) verbs)
(defprop problem (didn't-know misunderstood forgot) verbs)
(defprop change-of-mind (gave-up changed-his-mind) verbs)

(strip-expansions deep1)
((JOHN DECIDED-TO-HAVE A VACATION)
 (JOHN WENT-TO-BUY A TICKET AT A TRAVEL AGENCY)
 (BUT JOHN FORGOT JOHN'S WALLET)
 (JOHN GAVE-UP))

(find-expansions deep1)
((JOHN WAS VERY TIRED BECAUSE JOHN HADN'T HAD ANY BREAK FOR MONTHS)
 (JOHN DIDN'T USED TO CARE ABOUT MONEY)
 (JOHN WAS TOO LAZY TO GO BACK AND GET THE MONEY FOR THE TICKET))

; each sequence is summarized as a protagonist and a simple plot-unit
(summarize deep1)
((JOHN MOTIVATION) (JOHN EFFORT) (JOHN PROBLEM) (JOHN CHANGE-OF-MIND))

; read the list of verbs for the second text
(load 'semanticsd2)

(setq *protagonists* '(john))

(defprop motivation (was-tired) verbs)
(defprop effort (decided-to-have went-to-buy) verbs)
(defprop problem (forgot was-too-lazy) verbs)
(defprop change-of-mind (gave-up) verbs)

(strip-expansions deep2)
((JOHN WAS-TIRED)
```



```
(SO JOHN DECIDED-TO-HAVE A VACATION)
(JOHN WENT-TO-BUY A TICKET AT A TRAVEL AGENCY)
(JOHN FORGOT JOHN+S WALLET)
(JOHN WAS-TOO-LAZY TO GO BACK)
(SO JOHN GAVE-UP))
```

```
(find-expansions deep2)
((JOHN HADN+T HAD ANY BREAK FOR MONTHS)
 (JOHN DIDN+T USED TO CARE ABOUT MONEY)
 (GET THE MONEY FOR THE TICKET))
```

; the second summary is different from the first

```
(summarize deep2)
((JOHN MOTIVATION)
 (JOHN EFFORT)
 (JOHN EFFORT)
 (JOHN PROBLEM)
 (JOHN PROBLEM)
 (JOHN CHANGE-OF-MIND))
```

; read the list of verbs for the third text

```
(load 'semanticd3)
```

```
(setq *protagonists* '(john))
```

```
(defprop failure (hadn+T-had didn+T-used-to-care) verbs)
(defprop problem (was-very-tired forgot was-too-lazy) verbs)
(defprop effort (decided-to-have went-to-buy) verbs)
(defprop change-of-mind (gave-up) verbs)
```

```
(strip-expansions deep3)
((JOHN HADN+T-HAD ANY BREAK FOR MONTHS)
 (SO JOHN WAS-TIRED)
 (SO JOHN DECIDED-TO-HAVE A VACATION)
 (JOHN DIDN+T-USED-TO-CARE ABOUT MONEY)
 (SO JOHN FORGOT JOHN+S WALLET)
 (JOHN WENT-TO-BUY A TICKET AT A TRAVEL AGENCY)
 (JOHN WAS-TOO-LAZY TO GO BACK)
 (SO JOHN GAVE-UP))
```

```
(find-expansions deep3)
((GET THE MONEY FOR THE TICKET))
```

; the third summary is different from both the first and the second

```
(summarize deep3)
((JOHN FAILURE)
 (JOHN MOTIVATION)
 (JOHN EFFORT)
 (JOHN FAILURE))
```

(JOHN PROBLEM)

(JOHN EFFORT)

(JOHN PROBLEM)

(JOHN CHANGE-OF-MIND))

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

```
;; EXAMPLE 3: the CONDOMINIUM text
```

```
;; This example shows how different forms of analogy can be applied to texts.
```

```
; read the texts
```

```
(load 'condo)
```

```
(setq text1
```

```
'(0
```

```
(Paul wants to buy a condominium  
because Paul is sick of Paul's old apartment)  
(But Paul does-not-have money to buy a condominium  
because Paul bought a Porsche)  
(So Paul decides-to-give-up  
because Paul does not want to sell the Porsche  
and have the money back)))
```

```
(setq text2
```

```
'(0
```

```
(Paul wants to buy a condominium  
because Paul is sick of Paul's old apartment)  
(But Paul does-not-have money to buy a condominium  
because Paul bought a Porsche)  
(So Paul decides-to-borrow some money from a friend  
because Paul does not want to sell the Porsche  
and have the money back)))
```

```
(setq text3
```

```
'(0
```

```
(Paul wants to buy a condominium  
because Paul does not have a condominium)  
(But Paul does-not-have money to buy a condominium  
because Paul lost Paul's job)  
(So Paul decides-to-give-up  
because Paul does not want to borrow any money)))
```

```
; read the list of verbs which are associated with and identify simple
```

```
; plot-units
```

```
(load 'semantics)
```

```
(defprop failure (wants needs fails misses is-willing-to tries-to) verbs)
```

```
(defprop problem (does-not-have cannot cannot-afford cannot-reach cannot-do)  
verbs)
```

```
(defprop change-of-mind (decides-to-give-up changes-his-mind  
does-not-try-anymore gives-up thinks-it-is-not-the-case  
thinks-it-is-impossible) verbs)
```

```
(defprop effort
```

```
(decides-to-do decides-to-have decides-to-get decides-to-borrow  
decides-to-try decides-to-put-energy-into
```

decides-to-make-some-effort) verbs)

```
(setq *protagonists* '(paul))
```

```
(strip-expansions text1)
((PAUL WANTS TO BUY A CONDOMINIUM)
 (BUT PAUL DOES-NOT-HAVE MONEY TO BUY A CONDOMINIUM)
 (SO PAUL DECIDES-TO-GIVE-UP))
```

```
(find-expansions text1)
((PAUL IS SICK OF PAUL+S OLD APARTMENT)
 (PAUL BOUGHT A PORSCHE)
 (PAUL DOES NOT WANT TO SELL THE PORSCHE AND HAVE THE MONEY BACK))
```

```
(setq summary1 (summarize text1))
((PAUL FAILURE) (PAUL PROBLEM) (PAUL CHANGE-OF-MIND))
```

```
(setq links1 (find-links summary1))
((PAUL FAILURE < PROBLEM) (PAUL PROBLEM < CHANGE-OF-MIND))
```

```
(setq complex1 (find-complex-plot-units links1))
((PAUL GIVING-UP))
```

```
(strip-expansions text2)
((PAUL WANTS TO BUY A CONDOMINIUM)
 (BUT PAUL DOES-NOT-HAVE MONEY TO BUY A CONDOMINIUM)
 (SO PAUL DECIDES-TO-BORROW SOME MONEY FROM A FRIEND))
```

```
(find-expansions text2)
((PAUL IS SICK OF PAUL+S OLD APARTMENT)
 (PAUL BOUGHT A PORSCHE)
 (PAUL DOES NOT WANT TO SELL THE PORSCHE AND HAVE THE MONEY BACK))
```

```
(setq summary2 (summarize text2))
((PAUL FAILURE) (PAUL PROBLEM) (PAUL EFFORT))
```

```
(setq links2 (find-links summary2))
((PAUL FAILURE < PROBLEM) (PAUL PROBLEM < EFFORT))
```

```
(setq complex2 (find-complex-plot-units links2))
((PAUL WAY-OUT))
```

```
(strip-expansions text3)
((PAUL WANTS TO BUY A CONDOMINIUM)
 (BUT PAUL DOES-NOT-HAVE MONEY TO BUY A CONDOMINIUM)
 (SO PAUL DECIDES-TO-GIVE-UP))
```

```
(find-expansions text3)
((PAUL DOES NOT HAVE A CONDOMINIUM))
```

```
(PAUL LOST PAUL'S JOB)
(PAUL DOES NOT WANT TO BORROW ANY MONEY))
```

```
(setq summary3 (summarize text3))
((PAUL FAILURE) (PAUL PROBLEM) (PAUL CHANGE-OF-MIND))
```

```
(setq links3 (find-links summary3))
((PAUL FAILURE < PROBLEM) (PAUL PROBLEM < CHANGE-OF-MIND))
```

```
(setq complex3 (find-complex-plot-units links3))
((PAUL GIVING-UP))
```

```
(strongly-analogous? text1 text2)
NIL
```

```
(strongly-analogous? text1 text3)
NIL
```

```
(strongly-analogous? text2 text3)
NIL
```

```
(weakly-analogous? text1 text2)
NIL
```

```
; text1 and text3 have the same summary (plot-unit configuration)
(weakly-analogous? text1 text3)
T
```

```
(weakly-analogous? text2 text3)
NIL
```

```
; text1 and text2 have the same expansions
(similar? text1 text2)
T
```

```
(similar? text1 text3)
NIL
```

```
(similar? text2 text3)
NIL
```