8 Application-oriented
computational semantics

Sergei Nirenburg and Christine Defrise

8.1 Introduction

Large practical computational-linguistic applications, such as machine translation systems, require a large number of knowledge and processing modules to be put together in a single architecture and control environment. Comprehensive practical systems must have knowledge about speech situations, goal-directed communicative actions, rules of semantic and pragmatic inference over symbolic representations of discourse meanings and knowledge of syntactic and phonological/graphological properties of particular languages. Heuristic methods, extensive descriptive work on building world models, lexicons and grammars as well as a sound computational architecture are crucial to the success of this paradigm. In this paper we discuss some paradigmatic issues in building computer programs that understand and generate natural language. We then illustrate some of the foundations of our approach to practical computational linguistics by describing a language for representing text meaning and an approach to developing an ontological model of an intelligent agent. This approach has been tested in the DIONYSUS project at Carnegie Mellon University which involved designing and implementing a natural language understanding and generation system.

8.2 Semantics and Applications

Natural language processing projects at the Center for Machine Translation of Carnegie Mellon University are geared toward designing and building large computational applications involving most crucial strata of language phenomena (syntactic, semantic, pragmatic, rhetorical, etc.) as well as ma-
Application-oriented computational semantics

be task-driven — which means that adequacy and efficiency of description take precedence over generalization.

We also subscribe to a cognitivist approach to building NLP-related language processing theories, centred around the metaphor of the model of an intelligent agent. An NLP-related theory must account for such properties of intelligent agents as goal- and plan-directed activity, of which language activity is a part (verbal actions, together with perceptual, mental and physical actions, comprise the effector inventory of an intelligent agent). It must also take into account the knowledge of the agent’s attitudes to the entities in the world model as well as to remembered instances of events and objects in its own episodic memory. Not only are these attitudes often the subject of a discourse but they also influence the form of discourse on other topics.

8.3 Configuration

A semantic theory for natural language processing must address issues connected with the meaning-related activities in both natural language understanding and generation. While the semantic processing in these two tasks is different in nature (in lexical semantics, for instance, understanding centrally involves resolution of ambiguity while generation deals with resolution of synonymy for lexical selection), the knowledge bases, knowledge representation approaches and the underlying system architecture and control structures can be shared.

For instance, in the DIONYSUS natural language processing project many of the static knowledge sources and the basic control architecture are, indeed, shared, while the type of processing is quite different in different components of the system.1

In this section we describe the current computational configuration for implementing our agent-oriented theory of natural language processing. Not all facets of the agent model are implemented in our systems at this point, though we plan to remedy this situation. The current configuration of DIONYSUS contains the following modules:

1In fact, our general attitude to computational processing of natural language extends to other projects, such as TRANSLATOR (Nirenburg et al. [171]), POLAR (Nirenburg et al. [172]; see also Nirenburg and Lesser [98]), KBMT-88 (Goodman and Nirenburg [50]), OVTES (Monarch and Nirenburg [151]), and KIWI (Brown and Nirenburg [51]).
1. An ontology, a view of the intelligent agent’s world, including both
knowledge about types of things in the world and the so-called
‘remembered instances’ of these types; the ontology consists of

- a model of the physical world;
- a model of ‘self’, including knowledge of own goals and
  static attitudes to elements of the ontology and remem-
  bered instances of ontological objects;
- knowledge about the language communication situation.

2. A lexicon for each of the natural languages in the system. The lexi-
con contains the union of types of information required for analysis
and generation.2 The information in entries for polysemic lexical
items includes knowledge supporting lexical disambiguation. The
same type of information is used to resolve synonymy in lexical se-
lection during generation. A more detailed description of the struc-
ture of the lexicon in PROVOC is given in Meyer et al. [150].

3. A textual meaning representation formalism.

4. Knowledge about semantic processing, including

- structural mappings relating syntactic and semantic de-
  pendency structures;
- reference treatment rules (anaphora, deixis, ellipsis);
- unexpected input treatment rules (including metaphor and
  metonymy);
- text structure planning rules;
- knowledge about both representation (in analysis) and re-
  alization (in generation) of discourse and pragmatic phe-
  nomena, including cohesion, textual relations, producer at-
  titudes, etc.3

2In our earlier systems (see, e.g., Nirenburg and Raab [176]) we maintained that
different lexicons have to be produced for analysis and generation. It seems now that this
decision was in a large part induced by the logistics of knowledge acquisition in a large
application project. In fact, the overlap of the knowledge in the two lexicons is quite
considerable – even the collocation information that we once considered useful mostly
in the lexical selection process of natural language generation appears to be valuable in
certain situations in analysis.

3Some of the rules from this list can be physically a part of the lexicon; however,
they are mentioned overtly here because in most computational dictionaries this type of
knowledge is not used.

Text understanding consists in representing, using a specially designed no-
tation, the semantic and pragmatic (discourse, attitude, intention) informa-
tion encoded in each clause of a natural language input, augmented by the
representation of domain-related and text-related connections (relations)
among natural language clauses or sets thereof.

In fact, the final result of the process of text understanding includes
some information not overtly present in the source text. For instance, it may
include results of reasoning by the consumer, aimed at filling in elements
required in the representation but not directly obtainable from the source
text. It may also involve reconstructing the agenda of rhetorical goals and
plans of the producer active at the time of text production and connecting
its elements to chunks of meaning representation.

Early AI-related natural language understanding approaches were crit-
cised for not paying attention to the halting condition of meaning repre-
sentation. They were open to this criticism because they did not make a
very clear distinction between the information relayed in the text and in-
formation retrieved from the understander’s background knowledge about
the entities mentioned in the text. This state of affairs occurs when it is
decided that the program must apply all possible inferences to the results
of the initial representation of text meaning.

For a given ontology, we define text understanding as detecting and
representing a text component as an element of a script/plan (in Schank-
Abelson-Collingford-Wilensky’s terms — see Schank and Abelson [203],
Collingford [42], Wilensky [239]) or determining which producer goals are
furthered by the utterance of this text component. We stop the analysis
process when, relative to a given ontology, we find no more producer
goals/plans which can be furthered by uttering the sentence. But first we
extract the propositional meaning of an utterance using our knowledge
about selectional restrictions and collocations among lexical units. If some
semantic constraints are violated, we turn on metonymy, metaphor and
other ‘unexpected’ input treatment means. After the propositional mean-
ing is obtained, we actually proceed to determine the role of this utterance
in script/plan/goal processing. In doing so, we extract speech act informa-
tion, covert attitude meanings, and eventually irony, lying, etc.

There is a tempting belief among applied computational semanticists
that in an application such as MT, the halting condition on representing
the meaning of an input text can, in many cases, be less involved than the
general one. The reason for this belief is the observation that when a target
language text is generated from such a limited representation, one can in
many cases expect the consumer to understand it by completing the under-
standing process given only partial information. Unfortunately, since with-
out human involvement there is no way of knowing whether the complete
understanding is, in fact, recoverable by humans, it is, in the general case, impossible to posit a shallower (and hence more attainable) level of understanding. To stretch the point further, humans can indeed correctly guess the meaning of many ungrammatical, fragmentary and otherwise irregular texts (e.g. Charniak’s example of ‘lecture, student, confusion, question’ [28, page 150]). This, however, doesn’t mean that an automatic analyzer, without specially designed extensions, will be capable of assigning meanings to such fragments — their semantic complexity is of the same order as that of ‘regular’ text.

8.3.1 The concept of microtheories

Significant progress has been made recently in the field of computational linguistics with respect to the theories of syntax. Semantic and pragmatic phenomena have traditionally been less amenable to computational analysis. It does not seem plausible that an integrated semantic theory which would account for all lexical and compositional phenomena as well as the various pragmatic considerations will be formulated in the near future. This assessment becomes even more evident if one recognizes the necessity of providing heuristics for automatic recognition of the multiple meaning facets of natural language texts as a part of the theory. At the same time, linguistics has accumulated a significant body of knowledge about the various semantically laden phenomena in the natural languages (cf. Raskin [187] for a discussion of how this body of knowledge can be applied to computational analysis).

The above suggests that one of the more feasible ways toward building a comprehensive computational model of language understanding and generation behaviour in humans is to develop a large number of microtheories that deal with a particular linguistic phenomenon in a particular language or group of languages and then provide a computational architecture that allows the integration of the operation of all the modules based on these microtheories. Thus, one can envisage a microtheory of time, modality, speech act, causality, etc. In our work on DIONYSUS we have followed the microtheory-oriented methodology. While some of the microtheories we use in our systems are developed locally, many others are imported. The nature of the process of adapting a microtheory to the formalism and control conditions of a computational system is illustrated in Nirenburg and Pustejovsky [168] on the example of the microtheory of aspect.

To integrate the various microtheories in a single working system we suggest the use of a version of the blackboard computational architecture, in which a number of processes co-exist and, using a variety of background knowledge modules, collectively produce a desired output. The processors

are computational realizations of the various microtheories derived for the corresponding linguistic phenomena. These processors operate using data from the background knowledge repositories, such as grammars and dictionaries, as well as the intermediate results stored on the universally accessible set of blackboards. For a more detailed description of the model and its components see Nirenburg and Raskin [178], Nirenburg [169], and Nirenburg et al. [166].

In this paper we will discuss our theoretical position on the treatment of semantic and pragmatic phenomena in natural language processing environments. We will illustrate two of the components of a theory — the agent’s world model and the text meaning representation language.

8.4 Representation of agent’s knowledge

In formal semantics, one of the most widely accepted methodologies is that of model-theoretic semantics in which syntactically correct utterances in a language are given semantic interpretation in terms of truth values with respect to a certain model (to Montague semantics, a ‘possible world’) of reality. Such models are in practice never constructed in detail but rather delineated through a typically underspecifying set of constraints and illustrated through typically formal or miniature examples.

In order to build large and useful natural language processing systems one has to go further and actually commit oneself to a detailed version of a ‘constructed reality’. In practice, interpreting the meanings of textual units is really feasible only in the presence of a detailed world model whose elements are linked (either directly or indirectly, individually or in combinations) to the various textual units by a set of meaning-end relations. World model elements will then be densely interconnected through a large set of well-defined ontological links, properties, which will enable the world modeler to build descriptions of complex objects and processes in a compositional fashion, using as few basic primitive concepts as possible. Constraints on world model elements and their cooccurrence will serve as heuristics on the cooccurrence of lexical and other meanings in the text, thus facilitating both natural language understanding and generation.

A theoretically sound model of the world, an ontology, provides uniform definitions of basic semantic categories (such as objects, event-types, relations, properties, episodes, and many more) that become the building blocks for descriptions of particular domains and the creation of machine-traceable lexicons for comprehensive natural language processing. We believe that an optimum way of organizing this world model is as a multiply
interconnected network of ontological units, on which theoretically sound storage, access and update procedures will be developed.

An ontological model must define a large set of generally applicable categories for world description. Among the types of such categories are:

- Perceptual and common sense categories necessary for an intelligent agent to interact with, manipulate and refer to states of the outside world;
- Categories for encoding interagent knowledge which includes one's own as well as other agents' intentions, plans, actions and beliefs;
- Categories that help describe metaknowledge (i.e., knowledge about knowledge and its manipulation, including rules of behaviour and heuristics for constraining search spaces in various processor components);
- Means of encoding categories generated through the application of the above inference knowledge to the contents of an agent's world model (see articles in Brachman and Levesque [18]).

The choice of categories is not a straightforward task, as anyone who has tried realistic-scale world description knows all too well. Examples of the issues encountered in such an undertaking are illustrated in Gates et al. [63].

![Image](image_url)

Figure 8.1. A sample ontological subnetwork.

We have used the knowledge acquisition and maintenance system ONTOS (see Nirenburg et al. [167]) to produce several prototype ontological models. Figures 8.1 and 8.2 show several subnetworks from the ontology developed for and used in the KBMT-89 machine translation project (Goodman and Nirenburg [70]). These displays already illustrate answers to some of the above questions. The graphics browser of ONTOS facilitates fast overview and navigation in the ontological model. But this model is,

Application-oriented computational semantics

in fact, much more than a set of symbols (frame names) connected through is-a and part-of links.

Figure 8.3 illustrates the actual content of some of the nodes in this network.

![Image](image_url)

Figure 8.2. An event subnetwork.

The knowledge required in a model of an intelligent agent includes not only an ontological world model, as sketched above, but also records of past experiences, both actually perceived and reported. The lingua mentalis equivalent of a text is an episode, a unit of knowledge that encapsulates a particular experience of an intelligent agent, and which is typically represented as a temporally and causally ordered network of object and event instances.

The ontology and the episodes are sometimes discussed in terms of the contents of two different types of memory: semantic and episodic (e.g. Tulving [220]). This distinction seems useful in computational modelling as well. In our knowledge base we represent, with a varying degree of specificity, both ontological concepts and remembered instances of events and objects, which comprise the episodic memory.

Episodes are indexed through the type they correspond to and can be interrelated on temporal, causal and other links. The participant roles in the episodes can be either instantiations of object and event types in the semantic memory or references to existing named instances, stored outside semantic memory, but having links to their corresponding types (see figure 8.4). The figure illustrates the typology of structures comprising the world model of an intelligent agent. The basic ontological world model is augmented (for the purposes of specific processing types, such as analog-
ical reasoning) with a repository of the intelligent system's experiential knowledge. Our system must satisfy the knowledge representation needs of such a repository and abundantly cross-index it with the resident ontology. The presence of a systematic representation and indexing method for episodic knowledge is not only necessary for processing natural language but is also an enabling condition for case-based reasoning (Kolodner and Riebeck [126], Kolodner [125], Sclanik [202]) and analogical inference (e.g. Carbonell [25]).

A number of actual ontological decisions in the framework of the DIONYSUS project are described in Carlson and Nirenburg [26].

![Diagram of some frames relating to measurement.](image)

Figure 8.3. Some frames relating to measurement.

### 8.5 Representation of Text Meaning

The knowledge which the producer/consumer manipulates in order to successfully communicate includes the representations of various meanings of language elements, knowledge about the speech situation, including the knowledge about the interlocutor(s), knowledge about analysis and generation of the various language elements and knowledge about the world in general. Knowledge about the world is permanently stored in the ontology. Knowledge about treatment of language elements is stored in the various types of analysis and generation rules (syntactic, semantic, text planning, etc.). When actual processing occurs, structures are instantiated in the working memory of the intelligent agent which capture the knowledge necessary for the agent to 'understand' a text or to produce a text.

We believe that there is a well-defined set of knowledge elements whose existence constitutes a necessary and sufficient condition for a text to be considered 'understood'. In our theory the same elements are required to ensure successful generation of a text. The consumer understands messages communicated by the producer by 1) understanding and symbolically representing the meaning of the natural language text; 2) uncovering rhetorical relations among utterances and their components; 3) detecting the intentions of the producer in generating a discourse (in other words, those of the producer's plans and goals which are relevant to discourse); and 4) detecting the attitudes that the producer holds toward the content of the discourse. The producer, in creating discourse, goes through the following steps: 1) deciding what to say and elaborating a complete propositional meaning of the nascent text; 2) deciding on how to realize this meaning using realization means available in a natural language and paying attention to beliefs about the audience; and 3) taking into account the above and the preceding text (or dialogue), selecting and realizing an appropriate rhetorical structure of the nascent text.

Let us call the collection of knowledge elements sufficient for understanding or production of a text the text supermeaning. The supermeaning is a triad:

$$(8.1) \quad SM = T, G, S,$$

where $T$ stands for textual meaning, $G$ represents an agent's active set of goals and plans and $S$, the setting of the communication situation, including the pragmatic factors, such as forcefulness of expression, and parameters of the speech situation, including the spatiotemporal context and the properties of the participants and their relative social status, etc. In what follows we will discuss each of these elements in turn.
8.5.1 Textual meaning

The meaning of a text is a quadruple

$$T = (C, R, A, I),$$

where $C = \{\text{clause}_1, \text{clause}_2, \ldots, \text{clause}_n\}$ is a set of representations of the meaning of natural language clauses and $R = \{\text{relation}_1, \text{relation}_2, \ldots, \text{relation}_m\}$ is a set of connections among elements of the representation (clause components, clauses, sets of clauses). $A = \{\text{attitude}_1, \text{attitude}_2, \ldots, \text{attitude}_k\}$ is the set of attitudes to the various components of the meaning representation on the part of a text consumer or producer or another intelligent agent (in the case of "reported attitudes"). $I = \{\text{producer-intention}_1, \text{producer-intention}_2, \ldots, \text{producer-intention}_m\}$ represents the speech act meanings in the text.

The meaning of a clause is a quadruple

$$\text{clause}_i = (\text{ontol}, \text{tr}_i, \text{aspect}_i, \text{time}_i),$$

where $\text{ontol}_i$ is an instance of a concept in the ontology, possibly with additional constraints added to its ontological properties due to contextual influences in the input text. The features of the representation language which we use for representing semantic content of natural language clauses are a modified version of the Interlingua knowledge representation language used in the RBMT-99 machine translation system. Basically, we represent the semantic content of natural language utterances by instantiating ontological entities or reasserting remembered instances of such entities that are found (with the help of a lexicon) to be the most closely semantically related to lexical units in the input. The tasks of a) lexical disambiguation of candidate readings of the input lexical items and b) construction of a semantic dependency structure are performed by the semantic analyzer. The creation of the $\text{ontol}$ structures in clause meaning representations and in producer-intention representations is the only process in our theory where the ontology and the text meaning interact. $\text{tr}_i$ is a pair $\{\text{name}_i, \text{role}_i\}$ where $\text{name}_i$ is the name of a thematic role (sometimes also called case role or semantic actant - their inventory is listed among the relation-type properties in the ontology) and where $\text{role}_i$ is, again, an instance of an ontological concept, possibly with property values modified due to contextual influences. $\text{aspect}_i$ represents the aspectual properties of $\text{clause}_i$ through three dimensions - its phase (beginning, continuation, end), duration (momentary or prolonged) and iteration (single or multiple). The $\text{time}_i$ component of $\text{clause}_i$ is used for representing absolute time references (relative times are represented through relations, see below).
the producer during the planning stage of natural language generation is a widely accepted approach (cf. Moore and Paris [153]; Hoey [91]; Allen and Litman [2]). The plans to which these pointers point are realized in the text only when it is decided to produce a direct speech act, in which case the realization usually involves generating a separate target language clause. In the framework of natural language understanding, the representations of producer goals and plans are constructed by the consumer. The success of this process serves as a major halting condition on the process of semantic (and pragmatic) interpretation.

Depending on the context and other parameters, the producer may decide, for instance, to produce 'I will return at 10' or 'I promise to return at 10'. In the latter case the decision is made to realize the speech act overtly. The mechanism for this is as follows: traversing the goal-and-plan hierachy, the producer gets to the relevant point in the agenda, which is the (primitive) plan promise (or threat, etc., as the case may be). Since the realization rules for speech acts prescribe their realization as first-person singular clauses with the lexical realizations of the names of appropriate speech plans (acts), the natural language clause 'I promise X' goes produced, and eventually X is expanded into the subordinate natural language clause 'to return at 10'.

8.5.3 Speech situation

The speech situation, S, is represented as the pair

\[ \text{speech-situation} = \{ \text{deictic-indices}, \text{pragmatic-factors} \} \]

Deictic indices include the time and the place of the utterance of a text and the speaker and the hearer. The pragmatic factors describe stylistic parameters of the speech situation.

The microtheory of pragmatic factors used in our approach is largely inspired by Hoey [91]. The set of pragmatic factors in our model includes formality, simplicity, force, directness and respect. Values of these factors are represented as points on the range from 0 (low) to 1 (high). The default value of pragmatic factors is 0.5.

The various knowledge representation systems in artificial intelligence have traditionally stressed representation of propositional meaning (see, for example, Schank [201], Soeva [213], Creasy and Pollard [43]). Brachman and Schmolze [10], and many others. In this paper we will describe those components of the text meaning representation which are less frequently included in standard meaning representations – speaker attitudes, pragmatic factors and rhetorical relations.

8.5.2 The goal and plan component

The G component of the text supermeaning is a collection of structures of goal and plan instances which, the text consumer believes, are active in the text producer memory during the production of this text. Since we are interested in language communication, we describe only the goals and plans that a) presuppose the situation with at least two cognitive agents and b) relate to rhetorical realizations of goals. These pointers are not directly realized, but serve as background knowledge during the realization process of other elements of input. Reasoning about goals and plans of
8.5.4 A MICROTHEORY OF RELATIONS

In our present model we first distinguish text relations, domain relations and relations between intentions and domain-related text components. In what follows we further develop this taxonomy, taking into account semantic and pragmatic phenomena. For some of them we provide an example of lexical realization.\(^4\)

**Domain relations**

Domain relations connect events, states and objects. We further subdivide domain relations into six main subtypes: causal, conjunction, alternation, coreference, temporal and spatial. Discussion of these groups and their further subdivision follows.

**Causal:** Causal relations describe a type of dependence among events, states and objects. We distinguish the following subtypes:

1. *Volitional* causal relations hold between a deliberate, intentional action of an intelligent agent and its consequence. ('John turned the ignition key and the engine started.')

2. *Non-volitional* causal relations hold between a non-intentional action or a state of an intelligent agent and its consequence. ('I fell and broke my leg.')

3. Relations of *reason* hold between an event or state and a deliberate, intentional action by an agent. Often (but not always) they are lexically realized in English through 'because', 'since' or 'for the reason that'. ('I am prepared to help him because he helped you.')</n
4. *Enablement.* An event enables another event or a state when it removes the obstacles that were preventing the latter from occurring. ('Because the weather has improved, we will go on a walk.' 'The plug was leaking, so the water escaped.')

5. Event or state A is a *purpose* for event or state B if A describes a goal which an intelligent agent tries to achieve by performing B. ('He will leave early to catch the plane.')

6. Event or state A is a *condition* for event or state B if A is a cause, reason, enablement or purpose of B and A is an event or a state which has not actually happened and is, thus, hypothetical. ('If I win in the lottery, I'll travel to Java.')

**Conjunction:** The relation of conjunction holds among adjacent elements in a text that can be seen as components of a larger textual element. We distinguish the following subgroups:

1. *Addition* is a type of conjunction in which one (or more) of the conjuncts are set apart from others, sometimes for rhetorical purposes. ('Playing this piece involves real musical talent as well as technique.')

2. *Enumeration* is a type of conjunction in which all of the conjuncts have equal status. ('Athos went to Paris, bought a horse, visited his cousin, and strolled along the Seine.')\(^5\)

3. *Adversative* relations connect conjuncts whose differences are stressed in the utterance. ('Playing this piece involves real musical talent, not only technique.')

4. Event or state A stands in a *concessive* relation to event or state B if B is typically not believed to be a result of A. Often introduced in English by 'even though'. ('Even though the brick hit the window, the glass didn’t break.')\(^6\)

5. Entity A stands in the relation of comparison to entity B if the speaker believes that A and B are in some sense similar. ('Peter walked around the kitchen table like a hungry wolf.')\(^7\)

**Alternation:** Relations of alternation are used in situations of choice, parallel to the logical connector *or.*

1. *inclusive* or ('If you are an ex-prisoner of war or handicapped, you are entitled to state benefits.')

2. *exclusive* or ('I'll either go to the seaside or visit Florence.')\(^8\)

---

\(^4\) A full study and description of all the possible realizations of each subtype is needed. At this point we still have only partial results, which will be extended in the future.

\(^5\) In Quirk et al. [186], enumeration and addition are subtypes of *and.*

\(^6\) The distinction between adversatives and concessives was found in Rudolph [196].

\(^7\) This relation is adapted from Warner [237]; however, in Warner it is not considered to belong to the conjunction class.

\(^8\) This relation has been adapted from Warner [237].
Coreference: The relation of coreference is established among textual references to an object, an event or a state. Thus, in the following example: "(George Bush) hosted (Gorbachev) at the White House; then (he) invited (him) to Camp David; next day (the Soviet President) left for Minnesota" coreference relations hold among similarly bracketed entities. This relation is similar to designation in Quirk et al. [186, page 360].

Temporal: Our current theory of temporal relations is deliberately simplified. In our application work we do not seem to require a finer grain size of description of temporal relations. When such a necessity occurs, we will further develop our microtheory of time.

1. at: two events happen at the same time (the events can be either momentary or prolonged). ("When the war began he was traveling.")

2. after: one event happens after another in time. ("The king abdicated. Later, he was reinstated.")

3. during: one event takes place after the beginning and before the end of another event. ("While the war was raging, they exchanged letters about bird-watching.")

Spatial: We distinguish the following spatial relations: in-front-of, left-of, above, in, on and around. Just as in the case of temporal relations, we do not, at this point, have a finer-grain microtheory of spatial relations. The development of such a theory is among the directions of future research.

Text relations

The text relations are relations among elements of text - sentences, enumeration items, paragraphs, etc. - rather than among events and/or objects described in the text. We distinguish three types of text relations, as follows:

1. The particular relation connects two textual elements (sentences, paragraphs, etc.) one of which is an example or a special case of the other. ("It is important that young children should see things and not merely read about them. For example, it is a valuable educational experience to take them on a trip to a farm." Quirk et al. [186, page 668]).

Application-oriented computational semantics

2. The two textual elements connected by the relation of reformulation have a similar speaker meaning expressed in different ways. ("Peter works too much. That is to say, he neglects his family.")

3. In the relation of conclusion, a text element serves as a marker of end of discourse (including summary) for the other. ("In short, ..."

In an earlier version of this taxonomy (see, e.g. Defrise and Nirenburg [44]), we included additional text 'pointer' relations as well as relations delimiting discourse structure boundaries. We believe now that such relations (for instance, meanings like the one which is typically realized in English by 'in the previous chapter') are devices for maintaining text cohesion and readability and that they representationally belong in the text plan, the structure which is produced from text meaning representation during the process of natural language generation. In the text planning rules for DIOGENES, for example, we have used textual-before as a relation that connects text plan elements. In general, we would like to use the following taxonomy of these text plan relations.

1. Text pointers:
   - textual-before ("In the previous chapter, we dealt with the question of anaphora.")
   - textual-at ("We give a further example on page 6.")

2. Discourse structure boundaries:
   - now ("We will now turn to the next point.")
   - pop ("Anyway, that's why I came back.")
   - end ("And that's the end of the story.")

Intention-domain relations

These relations connect the events described in the text with the intentions of the speaker.

1. Temporal intention-domain relations connect the time of speech with the time of the action or event expressed by the utterance. There are two subtypes of such relations:
   - intention-domain-during ("It is raining. 'John is a teacher.'")
   - intention-domain-after ("Paul went to Paris.")
2. Motivation; this relation connects a speech act to a clause expressing an event or an attitude. The content of the clause motivates the reason to perform the speech act (i.e., the reason to utter the clause whose illocutionary force corresponds to the speech act). ('Your mother didn’t come back last night. Because the mail is still in the mail-box.' ‘Can you meet me at the office before three? Because I have a meeting at four.’) The causal link expressed by because in the first example above does not express a causal link between two domain-related facts, namely between a) the mother not coming back and b) the mail being out. It rather means that the producer can perform the speech act of informing the consumer of fact a) having made an inference based on fact b). In the second example above, the domain fact ‘I have a meeting at 3’ is the reason for the speech act of request-information.

8.5.5 Attitudes

In this section we will illustrate the types of attitudes we distinguish in our current microtheory.

The end points on the scale of the epistemic attitude correspond to ‘speaker doesn’t believe that X’ and ‘speaker believes that X’. A middle point on this scale corresponds to ‘speaker believes that possibly X’. The following example illustrates lexical realizations of the epistemic attitude (grouped by attitude-value, whose value should be understood as the midpoint of a fuzzy interval on the scale).

1. Paul left. I know for sure Paul left. I believe without doubt that Paul left. It is true that Paul left.

0.9 Paul must have left. Most probably, Paul left.

0.8 Paul may have left. I’m prepared to believe that Paul left. Perhaps Paul left. I’m almost sure Paul left.

0.6 It is possible that Paul left. I would think Paul left. Chances are Paul left.

0.5 I don’t know whether Paul left (or not).

0.3 It is unlikely that Paul left. I doubt whether Paul left.

0 Paul didn’t leave. It is impossible for Paul to have left. I don’t know that Paul left. I don’t believe (at all) that Paul left. It is not true that Paul left. I know that Paul didn’t leave. I believe (without a doubt) that Paul didn’t leave.

Application-oriented computational semantics

In our representation we do not distinguish what is from what the agent knows or is certain about. ‘Objective’ reality, thus, doesn’t exist in the system. Facts and events belong to the ‘projected world’ (Jackendoff [95, page 28]), i.e., reality as perceived by an intelligent agent. The fact that something is or is not, happened or did not happen, bears the mark of the agent’s perception. Hence the epistemic attitude. Degrees of knowledge are identified with degrees of belief and degrees of certainty. If an agent knows something, he is certain about it and believes it. ‘Paul left’ = ‘I (the text producer) believe that Paul left’ = ‘I know that Paul left.’

Similarly, we feel that if someone says ‘Paul didn’t leave’, it really means (to the text consumer who interprets it) ‘The producer doesn’t believe at all that Paul left’ = ‘The producer doesn’t know that Paul left’ = ‘It is impossible for Paul to have left’ = ‘The producer doesn’t believe that Paul left’ = ‘It’s not true that Paul left’. Negation can be understood as an attitude towards the event ‘Paul left’. Hence our decision to collapse the representation of the parity of an utterance with the epistemic attitudes of the agent. Seeing negation as the realization of an agent’s attitude has further advantages. Some uses of negation (the ‘polemic’ use, in denials) as in the following example:

(8.8) Paul said he came to the party yesterday. But he didn’t come.

( I saw him downtown with his girlfriend. At the time of the party, he was . . . ).

demand an analysis that takes into account more than parity, contrasting explicitly different agents’ attitudes towards the same event (this is similar to Ducrot’s [60] ‘polyphony’). We can provide a good representation of the above dialogue using the ‘attributed-to’ slot of an epistemic attitude frame. This representation will include the representation of the meaning of the clause ‘Paul came to the party yesterday’ in a TAMERLAN clause, say, clause₁, and two epistemic attitude frames, as follows:

attitude₁

| type: | epistemic |
| value: | 1 |
| attributed-to: | Paul |
| scope: | clause₁ |

attitude₂

| type: | epistemic |
| value: | 0 |
| attributed-to: | *speaker* |
| scope: | clause₁ |
In generating spoken text, the fact that the representation contains opposite epistemic attitudes with similar scopes will be realized through marked intonation. In contrast, a text featuring a simple negation (not a denial of a previous assertion, but a simple negative assertion) will not be represented using two opposite-value epistemic attitudes with similar scope.

Furthermore, representing parity as an attitude gives rise to ‘formulas’ that elegantly translate certain semantic relations between sentences. For instance the synonymy of the natural language sentences ‘The book is not interesting’ and ‘The book is uninteresting’ is translated in terms of attitudes as follows:

\[
\begin{align*}
\text{attitude}_3 & \quad \text{type: epistemic} \\
& \quad \text{value: 0} \\
& \quad \text{attributed-to: A} \\
& \quad \text{scope: clause_2} \\
\text{attitude}_4 & \quad \text{type: evaluative} \\
& \quad \text{value: 1} \\
& \quad \text{attributed-to: A} \\
& \quad \text{scope: clause_2} \\
\end{align*}
\]

The scale of the deontic attitude goes from ‘speaker believes that the possessor of the attitude must X’ (value 1) to ‘speaker believes that the possessor of the attitude does not have to X’. The realization of the deontic attitude can be illustrated as follows:

1 I must go. I have to go
0.8-0.2 I ought to go. I’d better go. I should go. You may go.
0 I needn’t go.

The scale of the volition attitude goes from ‘The possessor of the attitude doesn’t desire that X’ (attitude value 0) to ‘the possessor of the attitude desires that X’ (attitude value 1). Some illustrations of the realization of the volition attitude:

1 I wish ... I want to ... I will ... I will gladly ...
0.8-0.2 I hesitate to ... It may be a good idea to ... I’m reluctant to ...
0 I’m unwilling to ... I refuse to ... I don’t want ...

The scale of the expectation attitude goes from ‘The possessor of the attitude doesn’t expect that X’ through ‘the possessor of the attitude somewhat expects that X’ to ‘the possessor of the attitude expects that X’. Some lexical realizations of the expectation attitude:

1 Not surprisingly ... As expected ... Of course ... Needless to say ...
0.8-0.2 Even (as in ‘Even Paul left’)
0 Surprisingly ... It couldn’t be expected ...

The evaluative and saliency attitudes can have in their scope not only clauses, relations or attitudes like the previous ones, but also objects and

\[
\begin{align*}
\text{clause_2}\text{ represents the meaning glossed as ‘this book’ (because complete sentences only express the attitude toward the book). Therefore,} \\
(8.9) & \quad \text{epistemic 0 = epistemic 1} \\
& \quad \text{evaluative 1 = evaluative 0} \\
\end{align*}
\]
properties. It is therefore difficult to give a limited and exhaustive set of examples of realizations.

Evaluative attitudes are held toward events, things, properties, and relations among. One can also evaluate another attitude. Evaluation goes from 'the best for the possessor of the attitude' (value 1) to 'the worst for the possessor of the attitude' (value 0). Depending on the value of the scope of this attitude, realizations will greatly vary and may include no lexical realization at all. If the scope is an event, adverbs like fortunately and unfortunately will be used. If the scope is the physical appearance of a person, the endpoints of the scale of evaluative attitude will be realized as 'attractive' and 'ugly', etc. The meaning representations for the following sentences will centrally feature evaluative attitudes: 'This is the best book I ever read.' 'Blue is my favorite colour.'

The value of the saliency attitude varies with the importance that the agent attaches to a text component. High saliency is attached to entities that the agent wants to be stressed. The saliency attitude plays an important role in selecting the syntactic structure of the target sentences and in lexical selection for generation. Thus it will influence the order of elements in a conjunction; it will be realized syntactically through topicalisation ('It is Paul who won the contest') and lexically through connective expressions such as 'last but not least' or 'most importantly'.

We allow for the possibility of adding other types of attitudes. We have deliberately limited the number of possible attitude types to only a few, based on the immediate needs of our application work. It is clear however that a more complex and complete microtheory of attitudes will have to be developed. In particular, the evaluative attitude is a first-cut simplification of what should become a complex set of categories enabling us to represent the meaning of evaluative verbs, adjectives and adverbs featuring scalar behaviour. When trying to represent the meaning of sentences containing adjectives such as interesting, important or boring, one realizes that to use a single evaluative scale ranging from good to bad to represent all three adjectives is vastly inadequate, especially in view of the problem of lexical selection during generation; it only suffices as a first approach to capture argumentative orientations, which, as discussed in Ansecombe and Ducrot [3], play an important role in distinguishing, for example, the reasons for using but rather than and as a connector between two propositions.  

9The evaluative attitude to 'the book' in the example above would, in fact, belong to this latter class. Its scope is a clause only because there is no predication in the sentence other than the attitudinal one. In a sentence like 'John read an interesting book' the attitude is clearly toward an object instance.

10The status of attitude markers differs in the contexts of natural language analysis and generation. During analysis the values of attitude are part of the world of the con-

McKeown and Elhadad [146] also treat argumentative scales and attitudinals in a generation environment. They, however, consider these phenomena as part of syntax, thus avoiding the need to add a special pragmatic component to their system. This decision is appropriate from the point of view of minimizing the changes in an existing generator due to the inclusion of attitude information. However, if compatibility were not an issue, introducing a separate component would be a more appropriate choice.

8.5.0 PRAGMATIC FACTORS

Our microtheory of pragmatic factors is a modified version of the set of 'rhetorical goals' from Hovy [91]. In our systems, however, the processing performed with the help of the pragmatic factors is different. Among other differences, treatment of pragmatic factor values becomes in our systems an integral part of the process of natural language analysis, not only natural language generation.

Formality

The higher the formality, the closer to a legal or formulaic language one gets. Differences in formality will explain the distinction between 'I hereby inform you that X' and simply 'X'; between 'thereby' and 'thus'; between 'solicitation' and 'request'; between the use of full names and titles and abbreviations, etc. The following strategies were used by Hovy [91, pages 113-114] to make a generation system produce formal text: generate long, complex sentences by subordinating them in relative clauses; use passive voice, use 'more complex' tenses such as the perfect tenses; avoid ellipt-sis; use long and formal phrases and formal words; nominalize verbs and adverbs, as in 'their flight trajectory circled the tree' instead of 'they flew round the tree'; do not refer directly to the interlocutors; avoid contractions such as 'can't' or slang or colloquial words and expressions.
Simplicity

Formal simplicity and high value are somewhat related, and the values (formality: < 0.5) and (simplicity: > 0.5) will be highly correlated and will often be found together in one representation. Nevertheless, they differ. One can use a complex style (simplicity: < 0.5) and be either formal (formality: > 0.5) or not (formality: < 0.5). Simplicity: > 0.5 means the text will contain simple and short sentences, easy words, that its structure will be straightforward (the order of sentences will be such as to avoid the use of complex tenses for instance).

Force

'Forceful text is straightforward, direct, and has momentum.' (Hovy [91, page 124]). The strategies that a producer must use to make his text forceful include:

- use of enhancers such as 'not only . . . but also' and intensifiers;
- use of direct reference to the interlocutors;
- producing short, simple sentences, by choosing not to link sentences together and by including at most one adverbial clause;
- use of forceful or simple, plain words and phrases;
- no explicit realization of opinions (propositional attitudes) (using 'X' rather than 'I think that X' or 'It seems that X').

Colour

This parameter (a mixture of Hovy's colour and floriency) affects lexical selection in generation. Colourful text contains unusual words, idiomatic expressions, a relatively large number of adjectival modifiers ('a fresh new shop', 'a cruel and pitiless man'), including adjectival clauses ('Mr. X, a cruel and pitiless man, . . .').

Directness

This pragmatic factor relates to the amount of direct reference to the interlocutors. Different values of this parameter will influence the decision whether to use second person or impersonal forms of verbs, to express propositional attitudes explicitly or not ('My opinion is that X' vs. 'X'), to use impersonal or personal noun phrases, etc. In addition, a high value on this pragmatic factor will lead to overt representations of speech acts ('I promise that I will finish this assignment by tomorrow').

Application-oriented computational semantics

Respect

The relative status of the participants greatly influences the form of texts. The scale of possible relative social relationships is in reality very complex, and varies greatly from culture to culture. In our model, we simplify matters by keeping only three possible values: high, neutral and low. 'High' means that the producer is socially or hierarchically superior to the consumer; 'neutral' means the producer and the consumer have equal status; 'low' means that the producer is socially or hierarchically inferior to the consumer. This parameter will influence the generation of speech acts such as orders, requests, promises ("Would you be so kind as to X" vs. 'Please, X'); 'Could you tell me the time, please' vs. 'What's the time?') or salutations ('Good morning, Mr. X' vs. 'Hi'). This pragmatic factor will play a very important role in treating languages with well-developed systems of politeness, such as Japanese. In fact, the ability to use this parameter will allow one to carry the criticism directed at meaning-based machine translation1 to the effect that such 'nuances' are untranslatable and unrepresentable in a general meaning representation framework.

8.5.7 An annotated example of text meaning representation

The meaning representation we discuss in this paper is used in the natural language analysis system DIANA and the natural language generation system DIAGENES. For details of application-related formalization and actual use of this representation scheme see Nirenburg and Defrise [164]. In this section we would like to illustrate our representation on a small-scale example. Let's take sentence (8.10) as the input text.

(8.10) The train has, unfortunately, left.

The meaning representation of this sentence in TAMERLAN follows. The goals and plans of the speaker cannot be inferred by the reader of (8.10) outside an actual speech situation and in the absence of a broader textual context.

1See, for instance, Nagao [102, page 7]: '... In the case of translation through a pivot language, since it is necessary to decide upon expressions in the pivot language which will be appropriate for translations in which all of the factors found in all of the languages are included - including linguistic expressions dependent upon sex, social status, degree of respect, and so on ... The pivot language must be extremely well constructed if it is to include such subtle factors.'
(8.11) \[ SM_1 = \{ T_1, S_1 \} \]

\[ T_1 = \{ \text{clause}, \text{relation}, \text{attitude}, \text{attitude}, \text{producer-intentions} \} \]

\[ \text{clause}_1 = \{ \text{head} = \text{change-location}, \text{theme} = \text{train}, \text{source} = S_1, \text{deictic-indices}.\text{place}, \text{aspect} = \text{aspect}_1, \text{time} = \text{time}_1 \} \]

The head of the clause is an instance of the ontological concept change-location with the further constraint to the effect that the source thematic role is bound to the place of speech (in the ontology the constraint on the source of change-location is that it can be bound to any ontological descendant of the concept of physical-object). No additional constraints were possible for the destination thematic role of change-location.

(8.12) \[ \text{aspect}_1 = \{ \text{phase} = \text{begin}, \text{duration} = \text{momentary}, \text{iteration} = 1 \} \]

Our microtheory of aspect is based on theoretical work of James Pustejovsky and is presented in Nirenburg and Pustejovsky [109]. In this example we treat the aspect of the English leave as inchoative and non-iterative. As to duration, we consider this action to be momentary at the grain size of temporal description that we consider adequate for representing the meaning of this word.

(8.13) \[ \text{attitude}_1 = \{ \text{type} = \text{evaluative}, \text{value} = 0, \text{scope} = \text{clause}_1, \text{head}, \text{attributed-to} = S_1, \text{deictic-indices}.\text{speaker} \} \]

(8.14) \[ \text{attitude}_2 = \{ \text{type} = \text{epistemic}, \text{value} = 1, \text{scope} = \text{clause}_1, \text{head}, \text{attributed-to} = S_1, \text{deictic-indices}.\text{speaker} \} \]

The speaker's evaluative attitude toward the content of the sentence is very negative. This attitude reflects the meaning of unfortunately. The epistemic attitude shows that the speaker is quite sure that the train, in fact, left. Note that if in (8.10) leave had been modified by most assuredly or a similar intensifier, this meaning would have been realized through a salience attitude, with the value 1, whose scope would include the above epistemic attitude. The times of the attitudes are not informative and therefore are left out of the representation.

(8.15) \[ \text{producer-intention} = \{ \text{onto} = \text{assertive-act}, \text{scope} = \text{clause}_1 \} \]

assertive-act in our current ontology is a descendant of illocutionary-act and a sibling of commissive-act, directive-act, expressive-act and negative-act.

(8.16) \[ \text{relation}_1 = \{ \text{relation-type} = \text{intention-domain-after} \}
\text{first} = \text{time}_0 \]
\text{second} = \text{time}_1 \]

\text{relation-value} < 0.3 \]

The above temporal relation connects the time of speech and the time of the event described in the input. The value of the relation is a measure of the relative temporal distance between the two events. In our example the value in the lower third of the value range means that the event occurred recently.

(8.17) \[ S_1 = \{ \text{deictic-indices}_1, \text{pragmatic-factors}_1 \}
\text{deictic-indices}_1 = \{ \text{time} = \text{time}_1 \}
\text{pragmatic-factors}_1 = \{ \text{formality} > 0.5, \text{force} > 0.5 \} \]

The use of unfortunately raises the value of formality from the default level of 0.5.

8.6 CONCLUSION

The main points of this paper have been as follows.

- Practical results and systems-oriented research progress in computational linguistics are possible only if semantic analysis is based on a model of an intelligent agent, centrally including a world model, or ontology, a language for representing text meaning and a set of heuristic procedures for assigning meanings to natural language texts. The acquisition and representation of knowledge to support the above type of processing is a central problem of computational linguistics.

- Methodologically, the best approach to building comprehensive natural language processing systems seems to involve: a) producing a microtheory of every linguistic phenomenon that must be treated in a given application; these microtheories can either be developed from scratch or (and this is preferable) be adapted from treatments of various phenomena known from theoretical linguistics; b)
designing a control structure flexible enough to allow integration of all the procedures necessary for an application and reformulating the microtheories in terms of the formalism required by the control structure.

There are many unsolved problems in computational linguistics, and their complexity is typically very high. It is our belief that the field will be ill-served by approaches which stress deep and comprehensive treatment of one single phenomenon or a small group of phenomena. Such type of work leads to the production of microtheories. The latter are indispensable for the solution of the more general problems. But their availability alone does not constitute a sufficient condition for the achievement of practical computational-linguistic results. Microtheories of syntactic and semantic processing must be developed (these cannot possibly be derived from theoretical linguistics whose mandate does not include considerations of heuristic text processing). We need both the field work of creating world models and lexicons and computational environments to test the systems resulting from integrating microtheories and the software engineering substrate.

ACKNOWLEDGEMENTS

Members of the DIANA and DIOCENES projects have made important contributions to the research reported here. Parts of TAMERLANE have been suggested and/or improved by Boyan Onysikievych, Ted Gibson and Ralf Brown. Thanks are also due to Eric Nyberg, Lynn Carlson and John Leavitt. Ken Goodman did an excellent job criticizing the theoretical and methodological components of this paper. The authors are responsible for any remaining inconsistencies and deficiencies.

APPENDIX: A FORMAL DEFINITION OF THE TAMERLAN LANGUAGE

<text> ::= <clause>+ <relation>* <attitude>* <producer-intention>*
<clause> ::= <head> <aspect> <time>
<head> ::= <is-token-of> <thematic-role>* <property>*
<is-token-of> ::= ONTOSUBTREE-OR(all)
; this function returns a DISJUNCTIVE SET of all the elements in the ontological hierarchy rooted at its argument(s)
<thematic-role> ::= <th-role-name> <value>
<th-role-name> ::= ONTOSUBTREE-OR(thematic-role)
; this list includes AGENT, DESTINATION, THEME, BENEFICIARY, EXPERIENCER, etc.
<value> ::= <individual> | <collection>
<individual> ::= ONTOSUBTREE-OR(all) | <concept-instance> | <timed-concept-instance> | <symbolic-set-value> | <numerical-value> | <special-value>
<concept-instance> ::= TOKEN(ONTOSUBTREE-OR(all))
; TOKEN is a function that creates a token of its argument(s) which must be an ontological concept or a set of such
<timed-concept-instance> ::= <concept-instance> <instance-time>
; timed-concept-instances are used to refer to instances whose properties BUT NOT
IDENTITY changed with time (my car before
I painted it red and my car after I
painted it red are two timed instances of
the instance "my car")

\<instance-time> ::= since \<time\> until \<time\> | since \<time\> until \<time\>

\<symbolic-set-value> ::= a string from the slot "range" of
ONTOSTORE-OR(\text{attribute})

\<numerical-value> ::= \<point\>* | \<semi-interval\>* | \<interval\>*

\<semi-interval> ::= \<point\> \<point\> |
\<interval\> ::= \<point\> \<point\>

\<point\> ::= a number between 0 and 1, inclusive

\<special-value> ::= \*\text{producer}\* | \*\text{consumer}\*

\<collection\> ::= (UNION \<collection\>\<collection\>) |
INTERSECTION \<collection\> \<collection\> | |
SET-DIFFERENCE \<collection\> \<collection\> |
\{element\} \<collection-type\> \{\text{cardinality}\}*

\<element\> ::= ONTOSTORE(\text{all})

\<collection-type\> ::= disjunctive | conjunctive

\<cardinality\> ::= integer

\<property\> ::= \<property-name\> \<value\>

\<property-name\> ::= ONTOSTORE(\text{property})

\<aspect\> ::= phase duration iteration

\<phase\> ::= begin | continue | end

Application-oriented computational semantics

\<duration\> ::= momentary | prolonged

\<iteration\> ::= integer | multiple

\<time\> ::= \<absolute-time\> | \<time-variable\>

\<absolute-time\> ::= \text{e.g. 05:12-00-13:45:11.56}

\<time-variable\> ::= \text{time.\text{integer}}

\<relation\> ::= \<ordered-relation\> | \<unordered-relation\>

\<ordered-relation\> ::= \<relation-type\> \{\text{relation-value}\}
first \<relation-argument\>
second \<relation-argument\>
third \<relation-argument\>

\<unordered-relation\> ::= \<relation-type\> \{\text{relation-value}\}
arguments \<relation-argument\>
\<relation-argument\>*

\<relation-type\> ::= \<domain-relation\> | \<textual-relation\>

\<domain-relation\> ::= \<causal\> | \<conjunction\> | \<alternation\> | 
\text{attribute} | \text{coference} | \text{temporal}

\<causal\> ::= volitional | non-volitional | reason |
enablement | purpose | result | condition

\<conjunction\> ::= \text{simple-conjunction} | \<contrast\> | 
\text{comparison}

\text{simple-conjunction} ::= \text{addition} | \text{enumeration}

\<contrast\> ::= \text{adversative} | \text{concessive}
9 Form and content in semantics

Yorick Wilks

9.1 Introduction

This paper is written from the point of view of one who works in artificial intelligence (AI): the attempt to reproduce interesting and distinctive aspects of human behaviour with a computer, which, in my own case, means an interest in human language use.

There may seem little of immediate relevance to cognition or epistemology in that activity. And yet it hardly needs demonstration that AI, as an aspiration and in practice, has always been of interest to philosophers, even to those who may not accept the view that AI is, essentially, the pursuit of metaphysical goals by non-traditional means.

As to cognition in particular, it is also a commonplace nowadays, and at the basis of cognitive science, that the structures underlying AI programs are a guide to psychologists in their empirical investigations of cognition. That does not mean that AI researchers are in the business of cognition, nor that there is any direct inference from how a machine does a task, say translating a sentence from English to Chinese, to how a human does it. It is, however, suggestive, and may be the best intellectual model we currently have of how the task is done. So far, so well known and much discussed in the various literatures that make up cognitive science.

My first task in this paper concerns epistemology, but in a rather narrow way and does not directly address the large topics I have named above. It is to observe and criticise the fact that one school of AI researchers has, in effect, hijacked the word 'epistemology' and used it to mean something quite unrelated to its traditional meaning; the study of what we know and how we know it. The term has been used within the ongoing dispute in AI about how we represent knowledge (facts, generalizations, performances, etc.) in AI programs so that machines can be said to know things, or rather, how they can be programmed so as to perform as if they know things, such