Lecture Notes in Artificial Intelligence
Subseries of Lecture Notes in Computer Science
Edited by J. Siekmann

390

J.P. Martins  E.M. Morgado (Eds.)

EPIA 89
4th Portuguese Conference on Artificial Intelligence
Lisbon, Portugal, September 26–29, 1989
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong
The CASSIE Projects: An Approach to Natural Language Competence

Stuart C. Shapiro
Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-7022
U. S. A.
(716) 636-3182
shapiro@cs.buffalo.edu

1 Introduction

For a number of years, the SNePS Research Group at the State University of New York at Buffalo has been pursuing research on Knowledge Representation, Reasoning, and Natural Language Competence (NLC). Although a number of projects have been pursued, they have shared a common view of Intelligent Systems and of NLC. In this paper, I will present an overview of our approach to NLC, illustrated by some of the dialogues we have had with our systems. The group members whose work I will discuss include: William J. Rapaport, Janice Wiebe, Sandra L. Peters, Naicong Li, Soon Ae Chun, and Syed Ali.

Our approach to the general goal of Artificial Intelligence (AI) research (which we take to be the computational understanding of the processes needed to produce human-level general intelligence) has been to pursue general NLC and the knowledge representation and reasoning techniques needed to support it. We have been impressed by the amount of knowledge and general competence people gain through instruction carried out in their native language and through reading. As a result, our approach to building intelligent systems has not been to figure out how to program a computer in a programming language such as Lisp or Prolog, to perform some task or solve some problem in an intelligent way (other than the problem/task of general NLC or general reasoning), but rather how to produce a system that a person could instruct in NL how to perform the task or solve the problem. This is not to say that other approaches to AI are invalid, just that this is the approach we have decided to take.

Our work proceeds both theoretically and by building experimental NL interacting systems. These systems are all built on SNePS [19] as the knowledge representation system, SNIP [3, 6, 22] as the reasoning system, and a Generalized Augmented Transition Network (GATN) grammar [20] to specify the NL understanding and generation. As a result of the experiments and our developing theories, these underlying systems (which are implemented in Common Lisp) gradually change. To focus our thinking and our discussions, we have invented CASSIE, the Cognitive Agent of the SNePS System—an Intelligent Entity. CASSIE is the computational cognitive agent we interact with when we are interacting with one of these experimental NL systems. In a given system, a particular GATN parsing/ generation grammar specifies the SNePS arc labels and other representational constructs, that, in turn, represent CASSIE's beliefs (see [23] for more details). Since at any given time there are several group members working on different research issues, we tend to have several slightly different CASSIEs existing at once. That is why the title of this paper is "The CASSIE Projects," plural. Nevertheless, they are united by common underlying systems, philosophy, and architecture.

2 The CASSIE Architecture

The CASSIE system architecture is shown in Figure 1. Boxes show subsystems; circles show data structures; arrows show data flow. The human interlocutor types input to the GATN grammar, which, using the morphological analyzer and the lexicon to analyze words, parses the input and builds or accesses the belief structures stored in the SNePS semantic network, using SNePS commands and, where needed, SNIP. Responses to the human are formulated by the generation part of the GATN grammar using SNePS commands and, where needed, SNIP, to access information, and the morphological synthesizer
and the lexicon to formulate words.

Figure 2 is another view of the CASSIE architecture, focusing on a single interaction. An English statement or question is input; the GATN parser analyzes it in light of the current set of beliefs stored in SNePS; this may result in changes to the set of beliefs; and the SNePS structure that represents the main proposition of the statement or the answer to the question is given to the GATN generator, which outputs an English statement. Several points of this description are worth noting: although CASSIE produces an output sentence for each input sentence, and the GATN grammar has "sentence" as the highest-level syntactic structure, CASSIE is a discourse processor rather than a single-sentence processor, because each sentence is analyzed with respect to the belief structure built by previous sentences. The generator receives a proposition and uses the stored beliefs to help formulate the output sentence, especially to construct noun phrases. There is, as yet, minimal use of discourse or rhetorical structures in the planning and construction of output. These points will be illustrated further in the example interactions presented below.

3 Conversations with CASSIE

3.1 Basic Referring Exercises

The conversation of Figure 3 shows a basic interaction with CASSIE. The "-" character is CASSIE's prompt. The sentence after the prompt is input to CASSIE, and the following sentence is CASSIE's output. The number preceding the prompt has been added for this paper so we can refer to the interactions. This conversation is more fully discussed in [23], where the SNePS networks built as a result are also shown. SNePS networks will not be shown in this paper.

Before this conversation begins, the only thing in the SNePS network is a node representing the "current" time, indexed by the variable #0, which is accessible to the GATN parser. All words used in the conversation must already be in the lexicon, which includes the lexical categories noun, proper noun, pronoun, verb, auxiliary verb, adjective, determiner, and preposition. A lexical entry may have more than one lexical category—it will be disambiguated by the grammar. In particular, "pet" is stored as both a noun and a verb. The morphological analyzer determines that "petted" is the past tense of "pet".

As a result of analyzing sentence (1), CASSIE builds into the SNePS network structures representing the propositions: act(b1, pet, b2, t1, t2), Before(t1, t2),
1: Young Lucy petted a yellow dog.
   I understand that young Lucy petted a yellow dog

2: What is yellow?
   a dog is yellow

3: Dogs are animals.
   I understand that dogs are animals

4: Who petted an animal?
   young Lucy petted a yellow dog

5: Lucy is sweet.
   I understand that young Lucy is sweet

6: Lucy is a girl.
   I understand that sweet young Lucy is a girl

7: Girls are people.
   I understand that girls are people

8: What was petted by a person?
   a yellow dog was petted by sweet young Lucy

9: The dog is named Rover.
   I understand that Rover is a yellow dog

10: What was petted by a person?
    yellow Rover was petted by sweet young Lucy

Figure 3: Conversation showing reference to previously introduced entities.

Before(t2, NOW), Property(b1, young), Name(b1, Lucy), Property(b2, yellow),
Member(b2, dog). The proposition Act(b1, pet, b2, t1, t2) is passed to the generation part of the grammar as the main proposition of the input statement. The grammar outputs the canned phrase "I understand that" at the beginning of the response to every statement, then generates a statement that expresses the proposition it was given. Past tense is chosen for "pet" because the network contains the information that t2, the ending time of the event, is before NOW. The noun phrase generation sub-network retrieves all properties in the network that apply to the entity being described, and makes adjectives of them. It describes an entity by using its name if it has one; otherwise, it uses a class the entity is a member of. For a more detailed walk-through of a CASSIE grammar operating on a conversation very like that of Figure 3, see [20]. A shorter walk-through of sentence (1), explaining the SNIPS network it results in is presented in [16].

Interaction (2) shows CASSIE using the beliefs built in interaction (1) to answer a question. CASSIE says "a dog" rather than "the dog", because we have not yet implemented a facility for the proper creation of definite noun phrases.

Statement (3) is recognized by CASSIE as being an instance of a "bare-plural be bare-plural" sentence, and therefore it is understood as expressing the proposition Subclass(dog, animal). We believe that classification hierarchies can also be learned by intelligent systems through NL interaction, so such information is not pre-stored, but is built into CASSIE's belief structures as a result of such input.

Interaction (4) shows CASSIE answering a question based on the information of statement (1) and the class hierarchy learned in interaction (3). In this interaction, "dog" is used instead of "animal", because b2 is directly connected to dog in the network, but only indirectly connected to animal. A project that gets CASSIE to make the same choice for a better reason is described in [12] and [13].

With input (5), CASSIE learns an additional property that applies to Lucy. Although we realize that more than one person can have a particular proper name, and the representation used for CASSIE's beliefs are capable of storing a name as the proper name of more than one individual, CASSIE currently assumes that only one individual has a particular proper name (except for the case of nested beliefs—see Section 3.5). Therefore, CASSIE takes "Lucy" in statement (5) to refer to b1, the Lucy of statement (1). According to the noun phrase rules mentioned above, CASSIE would now describe b1 as "sweet young Lucy," as she does in output (6). However, that would make output (5) be "I understand that sweet young Lucy is sweet." This points out the necessity of keeping track of the clause-level proposition when noun phrases within the clause are being generated, so that the information of the clause is not also incorporated into the phrase.

Inputs (6) and (7) add a small class hierarchy above b1, which is tested by question (8). The topic-comment structure of question (8) was reflected in the answer by using the
same voice (active vs. passive) in the answer as was used in the question. (The lexicon explicitly lists "people" as the plural of "person").

In analyzing input (9), CASSIE examines her beliefs for dogs, and, finding exactly one, takes "the dog" to refer to b2. We do not yet handle the case where there are multiple possible referents for a definite noun phrase. The case where there is no referent for a definite noun phrase is illustrated in Section 3.3. Input (9) results in a node representing the proposition Name(b2, Rover) being added to the SNePS network. The CASSIE grammar is written so that a Name(x, n) proposition is expressed by a sentence of the form "n is ___________" where the blank is filled with a description of x that does not use the name n.

Interaction (10) demonstrates the noun phrase formation rule that prefers to use the name of an individual whenever one is available. Notice that outputs (1), (4), (8), and (10) were all generated from exactly the same SNePS node—the one representing the proposition, Act(b1, pet, b2, t1, t2). The differences in the surface forms were caused by: whether the sentence was generated as the response to a statement or the answer to a question; the voice of the question being answered; and the information in the SNePS network at the time the sentence was generated. (Thus demonstrating that CASSIE is a discourse processor.)

3.2 Pronouns

Figure 4 shows CASSIE resolving pronouns. Additional examples are in subsequent figures. The pronoun resolution strategy, designed and implemented by Nacong Li [5], depends on an ordered focus list, based on but different from Sidner's [24]. Each component of our focus list consists of: a SNePS node that represents the referent; the gender of the referent, as far as currently known (male, female, either, or neither), initialized from the lexicon; the number of the referent (singular or plural); a grammatical/semantic category (see below); and a weight. The grammatical/semantic category codes the previous occurrence of a mention of the referent in the discourse. In decreasing order of initial weight, these categories include: subject of an active clause; non-reflexive direct object; non-reflexive indirect object; agent of a passive clause; reflexive object; possessor in subject noun phrase; possessor in object noun phrase. The weight of a component starts out at the initial weight for the grammatical/semantic category of its first mention. It is

1: Bill is clever.
I understand that Bill is clever.

2: He dislikes John.
I understand that clever Bill dislikes John

3: He hates him.
I understand that clever Bill hates John

4: Bill saw a professor.
I understand that clever Bill saw a professor

5: She is smart.
I understand that a professor is smart

6: She likes him.
I understand that a smart professor likes clever Bill

Figure 4: Resolving pronouns

decreased as CASSIE reads subsequent sentences, and may be incremented at subsequent mentions. A component is removed from the focus list when its weight drops below a certain threshold.

When a pronoun is encountered, its gender and number are compared with components of the focus list according to the syntactic case of the pronoun. If the pronoun is nominative, it is compared with all components in the focus list. If it is reflexive, it is compared only with the subject of the current clause. If it is accusative, it is compared with the components in the focus list except for the subject of the current clause. If it is possesive, it is compared with the components in the focus list plus those in the current clause. The matching component with the highest weight is chosen.

In sentence (3) of Figure 4, Bill is chosen as the referent of "he" because, as a past subject he had a higher weight than John, a past object. Then John is chosen as the referent of "him" because, as an accusative pronoun, it cannot match the subject of the current clause.

The lexicon lists "professor" as either male or female, and "she" in sentence (5) cannot match Bill or John because they are male, so it matches the professor, whose component is then changed to female only. Thus, in sentence (6), "she" must match the professor, and "him" matches the higher weighted of Bill and John, which, by now, is Bill.
When CASSIE reads a definite noun phrase, such as "the cat," "the dog," or "the boy," she tries to retrieve an appropriate referent from memory. At input (2), this retrieval is successful, and "the cat" is taken to be the yellow cat from input (1) that John petted. At input (4), the retrieval is unsuccessful. Because of this, "the dog" is taken to be a generic reference, and the statement is interpreted as expressing the proposition \( \text{Subclass}(\text{dog}, \text{mammal}) \). The explanation of CASSIE's interpretation of statement (7) is a bit more involved, and depends on inputs (5) and (6).

Statement (5) expresses the rule that every dog is owned by a boy. We are currently working on more natural ways of expressing rules, but, as a temporary expedient, we have implemented parsing and generation grammars so that every rule that can be represented in SNePS can be expressed in a "formalized" English. It is a significant part of our approach to the general AI problem that we can use NL to give CASSIE rules that she can later use in reasoning.

Once CASSIE inputs statement (6), she has a dog "in mind," and the rule expressed by statement (5) implies that that dog is owned by a boy. When CASSIE tries to retrieve a referent for "the boy" in statement (7), backward inference activates the rule, which successfully fires, inferring a boy who owns the dog, and who is taken as the referent of "the boy." No special operation is needed for this. The same retrieval operation was performed to find a referent in inputs (2), (4), and (7). The difference is that in (2), an explicit referent (and no rule) was found, in (7) no explicit referent was found, but a rule was found that produced one, and in (4) neither an explicit referent nor a relevant rule was found.

Notice that inference is performed during the parse of the input, and that the interpretation of the sentence (whether specific or generic) can depend on the results of the inference.

Inputs (8) and (9) were given just to make interaction (10) more comprehensible to humans. Interaction (10) shows that CASSIE knows that the boy who saw Lucy is the owner of the dog Lucy petted. This information, the proposition that Bill owns Rover, was inferred during the analysis of sentence (7) (although CASSIE didn't know their names until sentences (8) and (9)); it only had to be retrieved to answer question (10).
1: John owns a cat.
   I understand that John owns a cat

2: He bought a canary.
   I understand that John bought a canary

3: The canary is named Tweety.
   I understand that Tweety is the canary

4: The cat stalks Tweety.
   I understand that the cat is stalking Tweety

5: His tail is swishing.
   I understand that the tail of the cat is swishing

6: His chirp alerted John.
   I understand that the chirp of Tweety alerted John

7: Lucy walked the dog.
   I understand that Lucy walked the dog.

8: The leash became tangled.
   I understand that the leash of the dog became tangled.

Figure 6: Implicitly introduced referents based on basic-level categories

3.4 Implicitly Introduced Referents

In the previous section, CASSIE reads a rule that every dog is owned by a boy, and uses this rule to infer the existence of a boy when a dog is mentioned. A more thorough treatment of implicitly introduced referents is reported in [14], from which Figure 6 is taken.

This figure is the only one in this paper that begins with a non-empty SNePS network. Instead, the network already has representations of basic-level categories, presumably acquired by children during perceptual interaction with members of the category. This information is represented as a large collection of SNePS default rules giving basic, general, default information about members of the category such as its normal parts, functional attributes, and thematic attributes. Because of the standard structure of these default rules, these parts and attributes can be retrieved from the node representing the basic-level category by following certain pre-specified paths in the SNePS network.

Whenever a basic-level category, or a member of a basic-level category is mentioned in CASSIE's inputs, the associated parts and attributes are placed in a secondary, potential focus list (secondary to the focus list used for pronoun resolution discussed in Section 3.2), along with the primary referent that activated them. If a subordinate-level category is encountered, the parts and attributes of its basic-level supercategory are used. We will say that a concept is activated when it is placed in the secondary focus list.

Whenever a definite noun phrase is encountered by CASSIE, the primary and secondary focus lists are searched before the general retrieval discussed in the previous section is carried out. If a match is found in the primary focus list, it is chosen as the referent of the definite noun phrase. If none is found there, but one is found in the secondary focus list, the definite noun phrase is taken to refer to a part or attribute of the basic-level referent stored with it in the secondary focus list.

When CASSIE reads "a cat" in sentence (1) of Figure 6, associated parts and attributes, including tail, are activated. When she reads "a canary" in sentence (2), the associated parts and attributes of bird, including chirp, are activated. Thus, "his tail" in sentence (5) is taken to be the tail of the cat, whereas "his chirp" in sentence (6) is taken to be the chirp of the canary.

Some concepts are activated by a combination of a basic-level category along with a particular context or event-type. The general parts and attributes of dogs are activated when CASSIE reads "dog" in sentence (7). Additional ones are activated by the event of walking the dog. These latter include leash, which enables "the leash" of sentence (8) to be recognized as the leash of the dog.

3.5 Nested Beliefs

Figure 7 shows a conversation with CASSIE involving nested beliefs. That is, at the end of the conversation, CASSIE has beliefs about other agents' beliefs. This conversation is based on work reported in [15, 17, 18, and 25].

Inputs (1), (2), and (3) introduce the characters sweet young Lucy and clever Bill. Sentence (4) is a report to CASSIE of a belief of John's. Interaction (5) makes it clear that CASSIE doesn't necessarily believe what she believes others to believe, while interaction (6) shows that she can answer "Who is properly?" questions.
1: Young Lucy is a girl.
   I understand that young Lucy is a girl.

2: She is sweet.
   I understand that young Lucy is sweet.

3: Bill is clever.
   I understand that Bill is clever.

4: John believes that Lucy is rich.
   I understand that John believes that Lucy is rich.

5: Who is rich?
   I don't know.

6: Who is sweet?
   Young Lucy is sweet.

7: John believes that Lucy is old.
   I understand that John believes that Lucy is old.

8: Lucy believes that Bill is stupid.
   I understand that Lucy believes that Bill is stupid.

9: Lucy believes that she is rich.
   I understand that Lucy believes that she is rich.

Figure 7: A conversation involving nested beliefs.

Output (7) shows that when constructing a noun phrase within a nested belief, CASSIE mentions properties she believes the believer (John in this case) believes to hold of the entity being described.3

It may not be as clear from the interactions, but CASSIE actually takes the Lucy of statements (4) and (7) to be a different Lucy from the one she believes to be sweet and young. This is at least a possible reading of them, so we are distinguishing the syntax of sentences like (4) and (7) from that of (8), where CASSIE takes “Bill” to refer to the Bill she already believes to be clever. Notice that "Lucy" in sentences (8) and (9) are taken to refer to sweet young Lucy.

---

3It is at least arguable that output (7) is standardly read to pragmatically imply that CASSIE believes Lucy to be rich. This needs to be pursued further.

---

1: Bill is Lucy's brother.
   I understand that Bill is Lucy's brother.

2: He is a professor.
   I understand that Bill is a professor.

3: Mary is his favorite student.
   I understand that Mary is Bill's favorite student.

4: Her dog is named Rover.
   I understand that Rover is Mary's dog.

5: John dislikes her dog.
   I understand that John dislikes Rover.

6: He said "her dog is ugly".
   I understand that John said "her dog is ugly", meaning Rover is ugly.

7: That John is narrowminded is Bill's favorite proposition.
   I understand that that John is narrowminded is Bill's favorite proposition.

8: Mary believes Bill's favorite proposition.
   I understand that Mary believes of John that he is narrowminded.

Figure 8: A conversation about a sentence and a proposition.

The "she" of sentence (9) is interpreted by CASSIE to be a quasi-indicator (see [15]), referring to Lucy, herself. That is the significance of the "*" in output (9).

In SNePS, propositions are represented as reified entities, and can serve as terms in other propositions. In particular, a nested belief is represented as a proposition that some agent believes another proposition. Propositions can be nested in this way arbitrarily deeply. CASSIE can also believe propositions to have properties. This is illustrated further in the next section.

3.6 Discussing Sentences and Propositions

Figure 8 illustrates that CASSIE can discuss propositions in other ways than as nested beliefs. It also shows that CASSIE can discuss sentences, and that it is important to retain a distinction between propositions and sentences. Soon Ae Chun wrote the sections of the CASSIE grammar illustrated in this conversation.
There is a table. The table is a support. Blocks are supports.

Before picking up a block the block must be clear. After picking up a block the block is not clear. If a block is on a support then after picking up the block the block is not on the support. If a block is on a support then after picking up the block the support is clear. After picking up a block the block is held.

Before putting a block on a support the block must be held. Before putting a block on a support the support must be clear. After putting a block on a support the block is not held. After putting a block on a support the block is on the support. After putting a block on another block the latter block is not clear.

A plan to achieve that a block is held is to pick up the block. A plan to achieve that a block is on a support is to put the block on the support. If a block is on a support then a plan to achieve that the support is clear is to pick up the block and then put the block on the table.

A plan to pile a block on another block on a third block is to put the third block on the table and then put the second block on the third block and then put the first block on the second block.

Figure 9: Blocks-world acting instructions.

After this instruction, CASSIE can be asked to perform various tasks in the blocks-world (which is simulated on a graphics screen). She uses these instructions to decide how to carry out the tasks, and to know what the state of the world is after each action.

The first paragraph of Figure 9 establishes the table, which is mentioned repeatedly in the instructions, and tells CASSIE what sorts of things are supports. The second paragraph gives effects and preconditions of picking up blocks. The third paragraph gives effects and preconditions of putting blocks on supports. The fourth paragraph gives some small plans, including one conditional plan. The last paragraph gives a longer plan for making a stack of three blocks. Notice that the plan is still sketchy—it must be filled in with clearing blocks, picking blocks, etc. from the preconditions of putting and the situation CASSIE finds herself in when she is asked to make a particular pile of three blocks.

3.7 Acting Instructions

A general AI system should not only interact in NL, and reason, it should also perform actions. (Actually, NL generation and reasoning are forms of acting.) Therefore, following our philosophy, we should be able to use NL to instruct an AI system how to act. Figure 9 shows part of a session in which CASSIE is instructed about the rules for action in a blocks-world. These inputs are shown in paragraph form, without CASSIE's output, but the actual session was run like the other CASSIE sessions, with CASSIE outputting a sentence after each input sentence.
Every sentence of Figure 9 except those in the first paragraph is interpreted as a rule to be stored in the SNePS network. These are examples of naturally expressed rules mentioned in Section 3.3. Notice that the variables of a rule are first introduced with an indefinite noun phrase that provides the restrictions on the variables. Later occurrences of a variable are in definite noun phrases, and may include discourse adjectives such as "latter" or "second."

More information on the project discussed in this section may be found in [4] and [21].

5 Conclusions

The CASSIE projects are united not only by shared computer systems and techniques, but by the philosophy that one powerful way to create intelligent systems is to create systems that can be instructed via natural language (and NL extended with graphics and gestures) what to believe, how to reason, and how to behave. All the versions of CASSIE analyze inputs with respect to stored beliefs that are modified by the inputs, and generate output based on current beliefs.

The dialogues illustrated and discussed in this paper should give the reader a feel for the CASSIE projects, and an idea of some of the NLC techniques we are using. More details may be found in the cited papers.

References


