Speech Communication 9 (1990) 565–585
North-Holland

LINGUISTIC PROCESSING RELATED TO SPEECH UNDERSTANDING IN SPICOS II

G.Th. NIEDERMAIR, M. STREIT and H. TROPF*
Siemens AG, ZFE IS KOM 31, Otto-Hahn-Ring 6, W-8000 Munich 83, Germany

Received 1 August 1990

Abstract. SPICOS II enables an on-going dialogue to take place with an office data-bank. This represents a further development of SPICOS I that, linguistically, was confined to a simple question–answer scheme. SPICOS II is speaker-adaptable and incorporates a vocabulary of some 1,200 words. The present article gives an overview of the elements connected with linguistic analysis. These elements are governed by a dialogue module that controls the dialogue and, by means of appropriate follow-up questions, avoids possible communications difficulties. The syntactic analysis is based on an augmented phrase-structure grammar. At the same time, semantic constraints are checked by means of a semantic network. Syntactic structure, together with semantic features and the results of the resolution of anaphoric bindings and a separate formal representation of discourse referents, is used to build formal-logic semantic representations of utterances. User presuppositions are also represented. From these formal representations date-bank queries are generated.

Zusammenfassung. SPICOS II ermöglicht einen Dialog mit einer Bürodatenbank. Dies stellt eine Weiterentwicklung von SPICOS I dar, das auf der linguistischen Seite nur ein einfaches Frage–Antwort-Schema unterstützte. SPICOS II ist sprecheradaptiv und umfaßt einen Wortschatz von ca. 1200 Wörtern. Der folgende Artikel gibt einen Überblick über die linguistischen Analysekomponenten. Diese werden von einem Dialogmodul gesteuert, das den Dialog leitet und mögliche Verständigungsschwierigkeiten zwischen Benutzer und System durch geeignete Nachfragen auffängt. Die syntaktische Analyse basiert auf einer erweiterten Phrasenstrukturgrammatik. Parallel dazu werden semantische Beschränkungen mithilfe eines semantischen Netzwerks überprüft. Die Bedeutung der Äußerung wird formallogisch dargestellt auf Basis der syntaktischen Struktur, der Ergebnisse der Anaphernresolution und einer von der semantischen Repräsentation getrennten formalen Repräsentation der Diskursreferenten. Die Präsuppositionen des Benutzers werden ebenfalls repräsentiert. Aus den formalen Darstellungen werden Datenbankabfragen generiert.

Résumé. SPICOS II rend possible un dialogue avec une base de données de bureau. C'est un prolongement de SPICOS I qui était limité, du point de vue linguistique, à un simple schéma de question/réponse. SPICOS II s'adapte au locuteur et possède un vocabulaire d'environ 1200 mots. Dans le présent article, nous donnons une vue d'ensemble du traitement linguistique de notre système. Un module de dialogue gère les composants linguistiques. Il contrôle le dialogue et évite, grâce à des questions adéquates, les problèmes de communications entre l'utilisateur et le système. L'analyse syntaxique se base sur une grammaire syntagmatique étendue. Parallèlement, un réseau sémantique vérifie les restrictions sémantiques. Dans le but de construire une représentation logique de la signification de la phrase, la structure syntaxique est utilisée en liaison avec les caractéristiques sémantiques, les résultats de la résolution anaphorique et une représentation formelle des référents du discours. Les présuppositions de l'utilisateur sont analysées et représentées. Ces représentations formelles peuvent être transformées en une interrogation de base de données.

Keywords. Natural language interface, speech understanding, linguistic processing, grammar, syntactic constraints, semantic constraints, formal semantic representation, discourse representation, dialogue handling.

1. Introduction

With SPICOS II (Siemens-Philips-IPO-Continuous Speech Understanding and Dialogue System) we have implemented a prototype which allows the retrieval of the contents of a database in

* The work described in this article has been carried out by G. Brockhoff, G. Dreckschmidt, G.Th. Niedermair, M. Streit, H.S. Tropf, U. Ziegenhain (Siemens-AG) and K. v. Deemter, J. de Vet (IPO, The Netherlands). The Project was funded by the German Ministry of Research and Technology under the number ITM 8801B9.

fluently spoken natural speech. SPICOS II works with a vocabulary of about 1,200 words and allows for questions concerning project members and communications, i.e. letters, articles, etc. The system rapidly adapts to new speakers.

The linguistic component consists of the specifications of the words in the lexicon, the syntax as an APSG (= Augmented Phrase Structure Grammar), a modified left-to-right top-down chart parser, a semantic network for the reduction of syntactically well-formed but semantically inadmissable structures, anaphora resolution, the generation of semantic representation in a typed logical language using lambda formalism, the separate representation of discourse referents and presuppositions, the generation of database queries, and the generation of an answer. A dialogue handler coordinates the different parts of the system and, in particular, allows backup questions from the system in order to give the user the opportunity to choose one of several sentences or meanings if more than one are generated.

In contrast to the SPICOS I prototype (Niedermair, 1987; Thurmair, 1986), which supports a simple question–answer scheme, the new system is able to maintain a dialogue where the user is able to refer back to previous utterances. In cases of ambiguity or error, the system will be able to question the user and clarify the problems in the ongoing dialogue. The linguistic components that analyse and interpret the output of the acoustic recognition will be dealt with in the following sections. They describe the stages in which the linguistic analysis is processed by the system and will explain their interdependence. Figure 1 shows an overview of the architecture of the whole system.

2. The dialogue-handler

With the exception of cooperative presupposition handling, which is described in Section 8, the following dialogue illustrates the capabilities of SPICOS II. It will also serve to explain the various steps of the dialogue and also the linguistic analysis components:

1 U(ser): Hat Herr Doktor Höge den Brief über Akustik an Herrn Noll geschickt?
(Has Doctor Höge sent the letter about acoustics to Mr Noll?)

1 S(ystem): Haben Sie gesagt: hat Herr Doktor

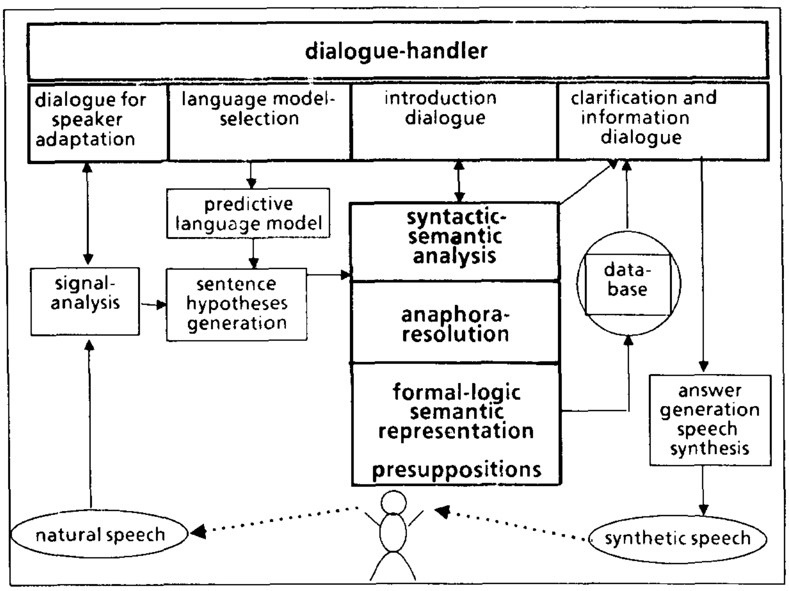

Fig. 1. System architecture SPICOS II.

Höge den Brief über Akustik an Herrn Noll geschickt?
(*Did you say: has Doctor Höge sent the letter about acoustics to Mr Noll?*)

2 U: Ja.
(*Yes.*)

2 S: Ja Höge hat den Brief geschickt.
(*Yes Höge has sent the letter.*)

3 U: War er auf dem letzten Treffen in Hamburg?
(*Was he at the last meeting in Hamburg?*)

3 S: Meinen Sie mit "er" Höge oder Noll?
(*By "he" do you mean Höge or Noll?*)

4 U: Höge.
(*Höge.*)

4 S: Nein Höge war nicht auf dem letzten Treffen.
(*No Höge was not at the last meeting.*)

Besides information requests and supplies, the above example contains stages involving clarification and verification (1S, 2U, 3S, 4U). The latter are the result of uncertainties in the system (1S) and from ambiguously resolvable situations like (3S). The permissible user reactions are either simple yes/no-answers to the system's questions or one-word replies (4U). If the recognition is uncertain, an echo-question (1S) is put by the system in order to obtain verification. (3S) is induced by a genuine ambiguity in the anaphora resolution mode. The dialogue handler takes the form of a finite state network. For questions (1U) and (3U) the system generates a formal representation of their meaning which is suitable for querying the database; for the other questions, the word content is passed on to the dialogue handler. The analysis for sentences like (1U) and (3U) comprises syntactic analysis, the analysis of semantic well-formedness in the application domain, the resolution of anaphoric references, the transformation into a formal logical representation, a semantic representation of anaphoric references and presuppositions and, finally, a transformation of the formal logical representation into a formal language suitable for evaluating the database. With the exception of the last phase, these components will be presented in the following chapters.

3. Syntactic processing

Typically, the syntactic component of a speech understanding system has two main tasks: firstly, it should reduce the number of hypotheses that emerge because of the relatively strong uncertainty of any acoustic recognition approach; secondly, it forms the basis for other linguistic components like semantic representation and anaphora resolution.

3.13.1. Acoustic-syntactic interface

Generally speaking, a grammar for normal, written input need not be as fine-grained as a grammar for the output following upon automatic speech recognition. In the case of written input, it can be assumed that normally the sentences are grammatically well-formed. So no more syntactic restrictions have to be considered than are necessary for an appropriate syntactic structure. However, when speech recognition output is processed, not only has an appropriate syntactic structure to be assigned, but all the syntactically ill-formed structures have to be excluded since, without higher knowledge sources, every word in the lexicon could theoretically be proposed at any point in time.

The acoustic component of SPICOS II outputs the acoustically best scored hypothesis for a continuously spoken sentence and, optionally, a number of suboptimal alternatives. Roughly speaking, the acoustic analysis is based on a stochastic bigram language model and the dynamic-programming algorithm for finding the most likely Hidden Markov Model state sequence(s) (Paeseler and Ney, 1989; Steinbiss, 1989). The bigram language model uses as its units not words, but categories. These categories are combinations of the syntactic and semantic categories that form part of the grammar and the semantic network.

With this approach, the task of the grammar is to reject proposed ill-formed sentences that cannot be excluded by *n*-grams. These are sequences where agreement conditions, discontinuous structures or long distance relations occur. In a sentence of more than two words where, for example, the verb is in initial and the subject in final position like in "Arbeitet hier heute Peter?" (literally: Works here today Peter?) the required number agreement between verb and subject cannot be captured with bigrams. So if more than one sentence hypothesis is proposed, the parser successively checks the top-*n* hypotheses until it meets a well-formed sentence.

3.2. Lexicon

The SPICOS II lexicon has full words, i.e. there is no separate morphological component in the grammar that deals with affixes and compounds, for example. A word form can belong to several different lexical categories (e.g. "arbeiten" (*work*): noun, verb). These categories are identical to the terminal categories in the syntax. Each category is associated with one or more feature bundles depending on the number of subclasses a word can belong to in its particular category.

A feature is composed of an attribute and a value, which can be either a single element or a set of elements. 17 different attributes are used. Element-valued features are "subclass", "capacity for passive", etc. Typical set-valued features are the morphological features that comprise the gender, number, case and declension for nouns, and the features relating to the subcategorization of verbs. Altogether about 13,000 value elements are assigned to attributes in the lexicon.

3.3. Format of the syntax

The SPICOS II syntax is an APSG and, in its present state, consists of about 100 main rules and 175 subrules, 12 terminal and 27 non-terminal categories are used. Every rule has

- a phrase structure part
  (e.g.: predicate → infinite-verb finite-verb);
- a feature checking part;
- a feature percolation part;
- one or more subrules, and
- an anaphoric part.

The feature-checking part contains conditions relating the feature of the right-hand-side categories. These conditions hold for all the corresponding subrules. The carry-over part assigns features to the left-hand-side category. They may be either newly created or are the result of operations on the right-hand-side features. This part also holds good for all the subrules of a main rule.

Within the feature-checking parts there are conditions that are operative concerning syntactic as well as semantic features. The latter describe the entities of the application domain or the relation to other entities (cf. Section 4). In the anaphoric part all the constituents are collected which are potential referents for anaphoric expressions (cf. Section 5).

Every subrule has

- a feature checking part, and
- a feature percolation part.

Contrary to the main rule, the parts of a subrule hold good for this subrule only.

Depending on whether a feature attribute takes as its value a single element or a set of elements, the following operations on features are possible: logical operators (not, and, or), equality, membership and subset predicate, and set operations (union, intersection, adjoining, difference). Set-valued features are used to keep the number of chart parser entries as small as possible. If no set-valued features were exploited, an extra entry would be necessary for each single element of a corresponding set. If all the conditions of the feature-checking part of a main rule and a subrule are fulfilled, the corresponding percolation parts are executed.

3.4. Concept of the syntax

In the SPICOS II syntax two kinds of rules are distinguished conceptually. The first concerns sentence topology (approximately "word order"). The categories of these rules are sentence functions, i.e. predicate; subject; direct, indirect and prepositional object; adverbial; and predicative. A direct object for example can be realized as a nominal phrase like "to see Peter", or as a subordinate clause like "to see that Peter is leaving". So, the second kind of rules determine how these functions can be realized in terms of syntactic constituents such as subordinate clauses and nominal and prepositional phrases.

The following fragment of syntax (without feature handling and subrules) is the basis for dealing with German sentence topology:

s → f fin-v f-plus-pred (1)

s → fin-v f-plus-pred (2)

f-plus-pred → f f-plus-pred (3)

f-plus-pred → pred (4)

("s" = sentence, "f" = function (excluding predicate), "fin-v" = finite verb, "pred" = predicate and "f-plus-pred" = function(s) together with the predicate.)

A realistic example for rule (1) is:

Wann schickte Peter seinen Artikel an Laura weg?
(*When did Peter send his article off to Laura!*)

and for rule (2):

Zeige mir den Artikel, den Peter an Laura schickte!
(*Show me the article that Peter sent to Laura!*)

The above rules allow for a rather free word order, which is necessary in German. Restrictions on this topology can be expressed by means of feature conditions, for example when several functions are realized as pronouns. To exclude ill-formed sentences as clearly as possible with respect to subcategorization, the recursive rule (3) employs feature checking to guarantee that no function apart from adverbial occurs more than once in the middle field, and that the functions are either obligatory or optional subcategories of the verb.

Assuming the underlying sentence-final position of the predicate, and with an appropriate mechanism in the chart parser to move the finite verb into this position, sentence topology in German can be captured with these rules, (cf. Section 4.3). The assumption concerning the final position of the predicate is induced by the final position of the following predicate elements: separated verb adjunct, infinite verb and verb in subordinate clause.

The only position of a function that cannot be analyzed by these rules is the relatively rare extraposition of a constituent to the right of the last element of a discontinuous predicate (e.g., "Hat Peter den Artikel geschickt *an Laura*?", literally: Has Peter the article sent *to Laura*?). However, an appropriate extension of the syntax is possible.

3.5. *Discontinuous structures*

An appropriate means for treating discontinuous structures is especially important for the predicate in German. As mentioned above, a finite verb may be separated from an adjunct like in "Wann *schickte* Peter den Brief *weg*?" (literally: When *sent* Peter the letter *away*?), or it may be separated from its infinite complement like in "*Hat* Peter den Brief *weggeschickt*?" (literally: *Has* Peter the letter *away-sent*?). For handling this kind of phenomena a mechanism called "shelf" is used in the chart parser. It is comparable with the hold register in an ATN parser (Bates, 1987). Similar concepts are proposed for a chart parser by Karttunen (1981) and Block and Haugeneder (1986).

When applying a rule that contains a category marked "moved", this category is "shelved" instead of being integrated into the parsed structure. During the succeeding parse the category is taken down and integrated into the new position as soon as a rule is applied that matches the category on the shelf.

As is indicated in the previous section, the finite part of a discontinuous predicate has to be moved into a sister position adjacent either to the infinite part, or to the verb adjunct at the end of the sentence. In principle, the part to be moved could be combined with several alternative complementary parts. In the sentence.

Hat Peter Briefe, die vom Mai sind, geschrieben?
(*Has Peter written any letters that are from May?*)

pred → fin-v (5)

pred → inf-v fin-v (6)

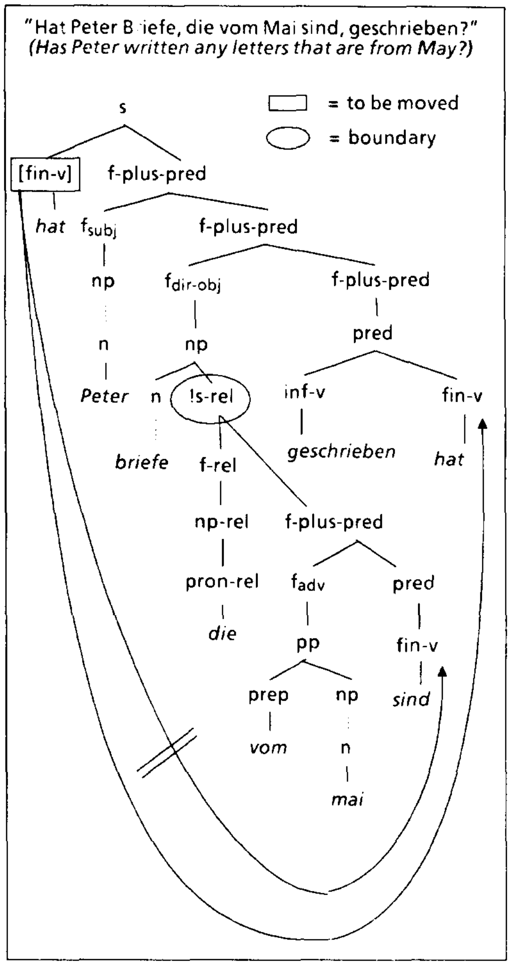

Fig. 2. Discontinuous structure with a moved element and a boundary rule.

the finite verb ("hat") can be moved into category "fin-v" under rule (5), which applies to relative clauses, or it can be moved into "fin-v" under rule (6), which applies to the main clause (cf. the two arrows in Figure 2). However, moving the finite verb into the predicate of main clauses is the only correct solution. To prevent such false movements certain rules have to be marked "boundary" (cf. the ellipse and the barred arrow in Figure 2). This means that no elements from outside can be consumed during the parsing of the corresponding category, in this case "pred" under "!s-rel". Movement within the scope of a boundary rule is still allowed, however.

3.6. Coverage

So far, the syntax covers

- mood: imperative sentences, yes/no-questions, wh-questions, declarative sentences;
- clause: main clauses, relative clauses;
- voice: active;
- tense: present, past, perfect;
- complex nominal and prepositional phrases, pro-forms for nominal and prepositional phrases, predicatives etc.

4. Semantic constraints

4.1. The different purposes of semantic feature restrictions

For continuous speech understanding systems it has now become an accepted fact that all available knowledge in the linguistic part of the process should be applied as soon as possible.

Semantic constraints limit the number of syntactic structures and so the search space for linguistic analysis. Since semantic restrictions are usually much more efficient than purely syntactic constraints in reducing the amount of possible combinations, these restrictions are applied at the earliest possible moment (Niedermair, 1986; Thurmair, 1988). This has often been achieved, especially in speech systems, by using semantic grammar variants (Mergel and Paeseler, 1987; Lowerre and Reddy, 1980; Schmandt and Arons 1986) or caseframe oriented approaches (Sagerer and Kummert, 1988; Poesio and Rullent, 1987; Shigenag et al., 1986; Young et al., 1988) with the known effects of either strong application dependence of the grammar or a loss of syntactic descriptive power in the case of caseframes if these are not used in combination with a syntactic grammar.

Caseframes also have clear limitations in expressing semantic relations other than "verb + sentence-function" or "noun + attribute". However, the semantic network not only permits the expression of semantic restrictions on role fillers in the most general way by means of a semantic hierarchy, but also allows the description of specification-relations (nominal or adjectival or other), attribute-value relations, term-classes (for resolution purposes), etc.

The early application of semantic restriction is achieved in SPICOS II by checking the semantic compatibility of content words. The tests on these constraints are included in the framework of the APSG (cf. Section 3.3). A semantic test forms part of a set of constraints which have to be met immediately each time the rule proceeds with a new constituent. The test accesses the semantic network and retrieves from it the semantic acceptability of a pair of content words together with additional information concerning the type of relation, e.g. the respective deep case in a noun–verb relation. This information on deep cases is stored in a separate feature (the deep case list) and is collected incrementally during parsing, so producing a full verb + deep case structure for the entire utterance. This structure is used by the formal semantic representation to identify the correct formal predicate-argument structure (cf. Section 6).

Apart from constraining the semantic content of an utterance the same semantic feature relations can be used in anaphora resolution for attaching semantic categories to semantically empty pronouns. The restrictions are also used for justifying anaphoric bindings, which are indirectly established through synonymy, hyperonymy or other relations. An account of the use of these relations in dialogues is given in Gehrke (1989). To model these semantic feature constraints, a semantic network has been set up which describes the semantic relations of the words in our application.

4.2. The structure of the semantic network

The network is a static model, not of the entities in the area of our application, but of the words describing these entities and their respective semantic relationships, i.e. the possible semantic behaviour of these words within an utterance. To avoid any confusion it should be pointed out that the network does not represent the data pertaining to the application as instances of its concepts, nor is it used as a means of representing meaning; this is achieved through other formalisms (cf. Section 6). A partial view of the semantic network is given in Figure 3.

Each content word is associated with a node (concept), of which it is an instance in a hierarchy of nodes. A node can, of course, have several instances if the words behave similarly in their relation to other words. Thus, the nodes are roughly equivalent to "semantic classes".

The basis of this network is a hierarchy of labelled nodes. The nodes in this taxonomy of classes are linked by "is-a"-relations. Words can, of course, also be instances of "higher" classes if a language provides words for more general concepts, such as "etwas (*something*)" that denote all objects except human beings, or "Dokumente (*documents*)" for all types of letters, protocols, etc. The requirement is that the word behaves like all the classes (and words) that it subsumes.

Besides the "is-a"-relation, a limited set of other relations is introduced that is necessary to express the various properties of the words' semantic behaviour. These include the following four relations:

"value-of": this links a concept to the nodes which describe the values that it can assume; e.g. the concept "person-name" is the value that the concept "person" (i.e. its instances) can assume. If in the application the persons bore numbers, the "person"-concept would be linked to numbers as well. This would allow us to analyse correctly "Mitarbeiter 714 (*colleague 714*)" as well as "Mitarbeiter Schmidt (*colleague Schmidt*)".

"role-of": this links a concept to the nodes whose instances we can describe as being roles associated with the concept. Thus "Mitarbeiter (*colleague*)" would be linked to "person" as being one of many roles that a "person" can assume, just as much as "Autor (*author*)" or "Absender (*sender*)". In a certain way they behave as their class-node "person" (e.g. in that they can be correctly linked as "agents" to action-nodes like "schreiben (*write*)". But they behave differently in other respects (e.g. "Absender der Firma (*sender of the company*)" is semantically incorrect (because in our application domain a "company" does not have a "sender"), in contrast to "Mitarbeiter der Firma (*colleague in the company*)", which we regard as semantically well formed.

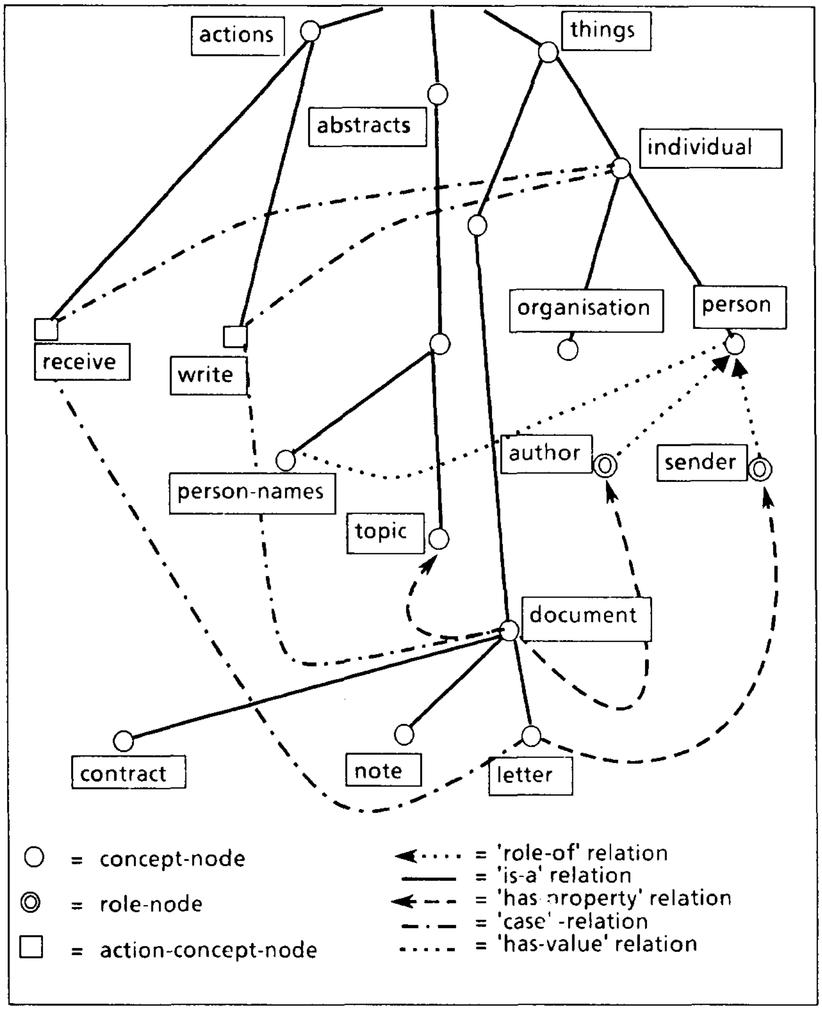

Fig. 3. Partial view of the semantic network.

"has-property': this links concepts to nodes whose instances can be regarded as their properties, e.g. there would be a link from "Brief (*letter*)" to "Absender (*sender*)" as a special property of "letter", which the "Brief (*letter*)" does not share with other documents. "Autor (*author*)" however would be linked to a higher node "Dokument (*document*)" since all subclasses of "Dokument (*document*)" can have an "author", including "letter". Therefore the property-links (as all other links) of "Dokument (*document*)" are assumed by the lower concepts.

"case-relation": this is a link from action-concepts, i.e. concepts usually expressed by verbs, to concepts that can act as fillers of a particular role (deep case) of this verb. This "filler-capability" is tied to additional restrictions like the preposition taken by a verb (in case of prepositional objects) or by a filler or both, or the function in sentence structure (e.g. direct object). These constraints therefore become additional arguments of the case-relation besides the two nodes. The deep case, as can be seen, is not as usual the label of the edge, but also an additional argument of it. This has the advantage that the rules used to traverse this edge need not be altered or supplemented with each new deep case. A shortcoming of this relation is that the combination of different deep cases cannot be expressed straightforwardly. Additionally, we do not create

instances of concepts during parsing. This would allow us to check which deep cases have already been filled for this concept and thus take into account certain restrictions in combination of deep case fillers; on the other hand it would lead to many partial instantiations of case frames during a parse, which we wanted to avoid because of efficiency considerations.

Only syntactic analysis is able to check through an appropriate feature whether a certain deep case has already been parsed or not, and thus to block multiple realisations of the same deep case for one verb.

Generally speaking, by means of this relation the network also contains maximally general frames for verb as well as for adjectives. The same holds for nominal–attribute relations, which are similar to verbs, but expressed by a different relation ("natrel") with fewer arguments.

As is usual in semantic networks, the lower level terms within our hierarchy implicitly inherit the definition (i.e. all the links of a node) of the higher nodes, and vice versa. This inheritance is not explicit and in our network it is effected through the transition rules only. This avoids this bidirectional inheritance assigning all the properties of a node to all the other nodes. The inheritance is realized between the specification of a general term (e.g. "document" can be specified by "topic" or "date") and its specialisations (e.g. "letter") and the specification of the specialised term (e.g. "letter" can be specified by "sender") and the more general term, but of course not its "sister-term". This allows a rather general description of, e.g. the constraints of verb frames, since a "case-relation" edge pointing to a concept can be satisfied by all the concepts that are subsumed by this concept. On the other hand, if a "case-relation" edge connects a verb concept to a more specialized concept, this "case-relation" may well be applied to the super-node of this concept. We want to avoid, however, that it can be applied to its sister nodes. We may want to specify, e.g. that "to be present" can be applied to "meetings", but not to "projects", which are sister nodes and subsumed under the concept "event". However, we still want to be able to speak of "being present at an event". Thus the more general term can inherit links from a more specialized node. Since this is only possible when the "event" concept is activated by having used the term "event" instead of "meeting", we avoid the "case-relation" link "be present" being automatically inherited by "project", which in this application is not desired. This is achieved by the transition rules. If they try to find a link (e.g. "be-present") for a concept (e.g. "event") they may also look at the subnodes and only "temporarily" inherit the "case-relation" link from there.

The "is-a" relation is also used for establishing the binding characteristics of words in anaphora resolution.

The leading criterion for the construction of the network is to find the most general description possible of the desired and semantically permissible relations in a given application domain (Chaffin and Hermann, 1984; Winston et al., 1987).

4.3. The transition rules

Semantic compatibility between words is defined in terms of rules which indicate which edges of the network may be traversed under certain circumstances of sentence surface structure in order to link two nodes. If two nodes can be linked through the application of these rules, its instances (the real words) are assumed to form a semantically correct relation. The transition rules for semantically correct relations differ depending on the sentence surface structure, in which the two words in question appear, like verb + noun, noun + specification (adjectival or prepositional), noun + noun, noun + noun-genitive, etc. Take for example a transition rule for two nodes, a noun and a verb. A semantic relation "semfit" between any verb "verbterm" and any noun "nounterm" is assumed to be correct if there is a link "case-relation" between "verbterm" and an "*n*-concept" whose instance or subclass-instance is "nounterm", and if "nounterm" is in a "role-of" relation to this concept or to any of its subclasses.

Since the network is imlemented in Prolog this can easily be stated as Prolog rules, such as:

semfit(_nounterm,_verbterm):-
case-relation(_n-concept,_verbterm,_preposit,_function,_case,_deepcase),
is-a(_n-lower,_n-concept),
role-of(_nounterm, _n-lower).

When checking a phrase structure in the syntactic grammar, these rules are applied to the relevant terms and other syntactic information in the phrase by means of a test operation. For this purpose a test-operator like "semfit" is introduced into the grammar, which takes the terms as its parameters. Depending on the syntactic categories of the words to be checked, different sets of transition rules are activated to prove the compatibility of the terms. In some cases, like the above mentioned verb-noun relations, the test can also return the deep case as a parameter, which is then stored as a grammatical feature and used to avoid multiple realisations of the same deep case in a sentence.

4.4. Use of the semantic network in anaphora resolution

The semantic network and its transition rules also serve to provide the module for anaphora resolution, with the proper semantic categories for anaphoric expressions in cases where the anaphor is realized through a pronoun, as in:

"schrieb er an den ... *(did he wrote to the ...)*".

The possible semantic categories for "er *(he)*" can be retrieved from the network by the proper application of the transition rules, (i.e. regarding the pronoun as a "variable" whose possible values have to be computed). In the above example, what would be retrieved would be the concept that the "case-relation" link from the concept-node of "schreiben *(write)*" immediately indicates as the subject. This is the concept "individual". Thus we would temporarily (i.e. for this sentence hypothesis) assign "er *(he)*" the semantic class "individual". In our office environment, and probably elsewhere as well, something of the class "individual" would be allowed, but not "event", for example.

We assign the possible semantic categories to a pronoun from its available sentential context. The possible categories for a pronoun are those which, in turn, would pass the semantic test for two content words as described above. However, this cannot always be done so easily with compound prepositional proforms like "daran", which contain the preposition "an" and a proform element "da(r)". The special treatment of these words will be discussed elsewhere.

In this way the class of the pronoun can be deduced from the network and returned to the syntactic analysis as the hypothetical semantic class of the semantically unmarked pronoun in this particular context. The anaphora resolution can use the instantiated semantic classe(es) of pronouns which have survived as part of correctly parsed sentence hypotheses in order to find correct antecedents within the range of these classes. This not only speeds up the search for possible antecedents, but also semantically constrains it.

The third purpose of the network becomes manifest when the resolution module tries to bind anaphoric expressions to their antecedents. In certain cases, where related terms like "Brief *(letter)*" and "Dokument *(document)*" are referentially bound, the resolution module needs to know any possible constraints on the semantic agreement between the two terms with respect to synonymy or hyperonomy, and other relations like part/whole, or property relations. The already mentioned hierarchy-relations and others like "property" and "role-of" in the semantic network provide this information. For the correct analysis and the correct binding of meronymic relations, an extension beyond the now existing "has-property"-relation is certainly required (Frederking and Gehrke, 1988; Gehrke, 1989). Similar to the semantic tests in the syntax, several sets of transition rules for various types of relations are used to state admissible relations between anaphora and antecedents.

5. Anaphora resolution

The task of syntactic anaphora resolution is to point to the possible antecedents of an anaphoric expression. Cases of unsolvable ambiguities are clarified in a subdialogue with the user (see (3S) in the dialogue example above). Resolution is achieved:

Firstly, by the collection of antecedent-candidates. In the simplest of cases these are nominal phrases, but they may also be larger phrases, or even full sentences. Our current application only requires the treatment of the first case. Together with an antecedent candidate, its relevant features taken from the syntactic analysis are stored in a workspace containing the dialogue history. Besides number and gender, such features include depth of embedding, recency, focus, topicalization, syntactic function, etc. However, it has been shown that a parallelism of syntactic functions mostly constitutes rather weak heuristic criteria for justifying anaphoric bindings in our type of dialogue. It would allow/demand a ranking of possible candidates in the case of weak evidence (together with an appropriate system response!), which we did not implement in the demonstration system.

Secondly, by the collection of anaphoric expressions. Currently we cover pronouns, nominal and prepositional proforms, and definite noun phrases. What syntactic elements are to be collected as possible candidates for antecedents as well as for anaphoric expression are encoded in the above-mentioned syntactic grammar. The percolation function in the grammar formalism is used to collect the relevant information for each candidate in the rule where it is processed. A special grammatical feature, the so-called "anaphora list", is used to store incrementally the relevant features for antecedent as well as for anaphora candidates. These anaphora lists are percolated to the sentence node so that the S-node also contains the complete list of candidates plus their relevant features, and this is transferred to the working area. The order of their appearance in the dialogue is only one feature among others.

Thirdly, by a reference grammar. This contains the rules that describe the conditions under which an anaphora candidate may be linked to an antecedent in the working area. A number of criteria are applied to the candidates such as syntactic agreement, recency of the antecedent in the dialogue, the fact of being mentioned repeatedly, and the fact of having already been taken up by a pronoun in the previous sentence. Since the conditions for associating an antecedent candidate with an anaphoric expression are rarely absolute, the rules are applied in succession so that the restrictions are "hardened" onto the possible sets of candidates. The process ends if

- no candidate can be found at all, or
- all rules are applied and more than one candidate remains, or
- only one candidate remains as possible antecedent.

The rules are applied in order of ascending restriction on the candidates. This ensures that from among the competing candidates those are dropped which are possible but less likely.

If there is an anaphoric expression with an appropriate antecedent, both are linked. This can lead to multiple links for one sentence, as is exemplified in dialogue (3U). The results of the resolution are therefore handed over to the dialoque handler, which disambiguates the reference by means of a clarification dialogue. The result of this disambiguation is returned to the anaphora resolution, which updates its links in consequence. These now unambiguous links are used in the following step by the formal logical representation to insert the proper representations of the antecedents in the place of the anaphoric expressions.

6. Semantic representation

In SPICOS the main topic of semantics is the representation of questions and certain kinds of imperatives. In the SPICOS system two types of questions can be distinguished: yes–no-questions and German w-questions. W-questions include English wh-questions and questions for the cardinality of set of entities (*wieviele* = *how many*). *Warum* (*why*) and *wie* (*how*) questions are excluded however. Yes–no-questions are expected to be answered with *yes* or *no*, so they can be represented by logical sentences which evaluate as true or false. W-questions are expected to evaluate to the entities that are asked for. They are represented as sets (or as functions from sets into integers in the case of *wieviel* (*how many*) questions). Imperatives are represented with the help of semantically empty predicates that trigger actions of the system as *side effects* (this is due to the fact that extensional logic is used). Other aspects of the user's questions, like the introduction of discourse referents and presuppositions, are considered in Sections 7 and 8.

Semantics in the SPICOS system works as a two-stage process. The first stage aims at a representation of meaning which is dependent of the domain of discourse. Basically, this means that the words of natural language function as the atoms of semantic representation. Of course this does not hold good for those words that contribute to the logical structure of an utterance, e.g. determiners.

On the one hand, by adopting this approach we free ourselves from special applications. But on the other, we transfer a lot of the ambiguity and vagueness of natural language into the formal representation of meaning.

If we want to deal with applications where knowledge is represented formally, we have to eliminate vagueness and make ambiguity explicit. A problem that is even worse is the following:

In applications, knowledge is modelled in different ways. Even if we concentrate on one type of application, say interfacing a relational database, we have to account for the fact that very different relations can be used to describe the same domain. But in dealing with different applications, we do not want to lose the advantage of being domain-independent. A solution to this problem is translation: The unspecific domain-independent representations are translated into the more specific domain-dependent ones.

We claim that it is possible to manage translation by translating the *atoms* (i.e. the non-logical constants) of domain-independent meaning representations into expressions that account for the special structure of the model that we want to deal with. In adopting this approach we stay in the tradition of the PHLIQA system (Bronnenberg et al., 1980) and the TENDUM system (Bunt et al., 1984).

In the first stage, an extensional type logic language is used as a representational language. This language includes set theoretical expressions. This feature is very useful for the representation of discourse referents (cf. Section 7) and presuppositions (cf. Section 8). In the SPICOS system we make a distinction between sets and predicates. Sets are of a set type, while predicates are of a function type (e.g. functions from sets to truth values). Set expressions are used to represent common noun phrases, predicates are used to represent verbs and relational nouns. Special constructions are added to handle time expressions and to describe parts of ordered sets.

In the second stage a sorted type language is used. While in the first stage we only consider one type of individual (*entity*), in the second the individuals are sorted. Both languages allow the use of lambda-expressions.

The languages are variants of the ELF/ELR (Ensemble Language Formal, Ensemble Language Referential) languages, which are extensively described in Bunt (1985). There, *Referential* is meant in the sense of domain-specific.

Translation is not a function, but a relation. To account for ambiguity, ELF constants can have more than one translation into ELR expressions. To account for vagueness, ELR translations of ELF constants can be (almost) arbitrarily complex. Translation is guided by a type reference mechanism, which strongly reduces combinational ambiguity.

In this article only the first stage of the semantic analysis will be considered. The referential representation and the translation process are not discussed.

SPICOS semantics, like semantics in TENDUM and PHLIQA, is an (extensional) descendent of Montague semantics. In the SPICOS I system, semantic rules work according to the general principles of Universal Grammar (Montague, 1970). For every context-free syntactic rule there is a semantic rule that determines the semantic representation of the left-hand side of the rule by the representations of the right-hand side operands of the rule. Basically, this is also the treatment in SPICOS II. For reasons which we will explain in Section 6.3, some deviations from the principle of compositionality are included in SPICOS II.

6.1. VP semantics

Consider utterance (1):

(1) Hat Schmidbauer einen Brief geschrieben?
(Has Schmidbauer written a letter?)

(1) is a yes–no-question. We aim at a description of the utterance as a closed logical formula as expressed in (2). SCHMIDBAUER is an individual constant, representing *Schmidbauer*. The common noun *Brief* is represented by the set constant BRIEF, the verb *schreiben* is represented by the two-placed predicate SCHREIBEN. The expression $x \in BRIEF$ means *x is a member of the set BRIEF.* We use the notation of restricted quantification. This means that (2) is equivalent to (2)′.

(2) $\exists x \in$ BRIEF : SCHREIBEN (SCHMIDBAUER, x)

(2)′ $\exists x$ (x $\in$ BRIEF & SCHREIBEN (SCHMIDBAUER, x))

The yes–no-question (1) contains two NPs, *Schmidbauer* and *einen Brief*, and the transitive verb *schreiben*. The NPs are *syntactic functions* (called f) in the sense of Section 3.

Syntax will first combine the NP (the f) *einen Brief* with the verb *schreiben* to form a so-called f-plus-pred, and will then combine the NP *Schmidbauer* with the f-plus-pred to form a sentence.

In parallel with the syntax, the semantics will first combine the representation of the NP *einen Brief* and the representation of the two-placed verb *schreiben*, which results in the one-placed predicate *einen Brief schreiben*. The semantics will combine the representation of *Schmidbauer* and the predicate *einen Brief schreiben* to form a logical sentence. We only consider the second step here.

Assume that we already have the one-placed predicate (3) as a representation of *einen Brief schreiben*.

(3) lambda y $\exists x \in$ BRIEF: (SCHREIBEN (y, x)

One-placed predicates denote functions from entities to truth values. (3) represents a function that can be applied to individuals. For example it can be applied to the constant SCHMIDBAUER.

(4) lambda y $\exists$ x $\in$ BRIEF: SCHREIBEN (y, x) (SCHMIDBAUER)

(4) leads after lambda-conversion to the desired result (2).

We could therefore associate semantic rule (5b) with syntactic rule (5a). F-PLUS-PRED denotes the semantic representation of f-plus-pred, F denotes the representation of f.

(5a) s $\rightarrow$ f f-plus-pred

(5b) *S* $\rightarrow$ *apply* F-PLUS-PRED *to* F

But this does not fit NPs that are not constants. To remain general we represent the NP *Schmidbauer* as a function working on predicates, as in (6).

(6) lambda P P(SCHMIDBAUER)

We can now apply (6) to the predicate (4) according to rule (7) instead of rule (5b).

(7) *S* $\rightarrow$ *apply* F *to* F-PLUS-PRED

As a result we get (8), from which lambda-conversion leads to the desired form (2).

(8) lambda P P(SCHMIDBAUER) (lambda y $\exists x \in$ BRIEF:SCHREIBEN (y, x))

Example (9) shows how sets are used to represent w-questions in the SPICOS system.

(9) Wer hat einen Brief geschrieben?
Who has written a letter?

We aim at a description of utterance (9) as a set, as it is expressed in (10).

(10) $\{y \in$ PERSONEN $\mid \exists x \in$ BRIEF: SCHREIBEN(y, x)$\}$

Usually NPs are considered as a function from predicates to sentences (i.e. truth values) or from predicates to predicates. In SPICOS we also allow NPs to denote functions from predicates to set expressions.

For example *wer* (*who*) has the representation (11).

(11) lambda P$\{y \in$ Personen $\mid$ P(y)$\}$

Rule (7) leads with (11) and (3) to the desired result (10).

6.2. NP semantics

Basically, NPs consist of determiners and common noun phrases. Roughly, the determiner determines the quantificational and referential behaviour of the NP. A common noun phrase determines the source of the quantification (Bunt, 1985) or the reference set (cf. Section 8) of the NP (Anaphoric relations are not considered in this section).

The representations of NPs are constructed by applying determiners to common noun phrases.

(12) Alle: lambda X lambda P $\forall x \in X$:P(x)
(all)
Mitarbeiter: MITARBEITER
(colleagues)
Alle Mitarbeiter: lambda X lambda P $\forall x \in X$:P(x) (MITARBEITER)
(all colleagues)
: lambda P $\forall x \in$ MITARBEITER:P(x)

(13) Welche: lambda X lambda P{$x \in X$ | P(x)}
(which)
Mitarbeiter: MITARBEITER
(colleagues)
Welche Mitarbeiter: lambda X lambda P{$x \in X$ | P(x)} (MITARBEITER)
(which colleages)
: lambda P{$x \in$ MITARBEITER | P(x)}

(12) shows an example of universal quantification, and (13) shows how the representation of WH-NPs are constructed. The last row in each example shows the result of the lamdba-conversion.

6.3. Deviations from compositionality

It seems very natural and unproblematic to represent the source (or reference set) of an NP as a closed set expression. Unfortunately, this leads to difficulties with the compositionality principle. Consider example (14), which contains an NP with a genitive-NP-attribute.

(14) Zeige die Briefe aller Mitarbeiter.
(Show the letters of all the colleagues.)

The use of the genitive introduces a relation between atribute and head that is expressed by the predicate VON (*of*). The semantic rules for the attachment of attributes cannot be realized in this paper. The reader may believe that we get (15) as a representation of the common noun phrase *Briefe aller Mitarbeiter* (*letters of all the colleagues*), if *alle* is represented as in (12).

(15) {$x \in$ BRIEFE | $\forall y \in$ MITARBEITER: VON (y,x)}

A letter is a member of the set (15) if it is shared by all colleagues. This interpretation is the most unlikely one. The problem could be solved by giving the attribute wide scope over the head of the NP and also over the verb. But this would lead to representation (16), which contains no set expression for the reference set of the NP.

(16) $\forall y \in$ MITARBEITER:ZEIGE({$x \in$ BRIEFE | : VON (y,x)})

To satisfy the condition that the source (reference set) of an NP should always be represented by a term, we have to represent the determiner *alle* in a different way. If *alle* determined existential quantification, representation (17) would be achieved, which would be the "source"-condition.

(17) {$x \in$ BRIEFE | $\exists y$ MITARBEITER: VON (y,x)}

It does not seem very attractive to give *alle* an existential interpretation lexically. The view, taken in SPICOS II, is that the syntactic context forces *alle* to be interpreted existentially. This means the representation of the embedded noun phrase *aller Mitarbeiter* is determined not at the level of the NP itself, but at a higher level.

7. Anaphora representation

One of the challenging problems of plural anaphora is the fact that they often cannot be represented by variables (i.e. set variables). Consider example (1).

(1) Haben alle Mitarbeiter einen Artikel veröffentlicht?
(Have all the colleagues published an article?)
Ja, alle Mitarbeiter haben einen Artikel veröffentlicht.
(Yes, all the colleagues have published an article.)
Zeige diese Artikel!
(Show these articles!)

The most likely interpretation of the first sentence of (1) is (2).

(2) $\forall x \in$ MITARBEITER: $\exists y \in$ ARTIKEL: VERÖFFENTLICHEN (x,y)

In the second question the user refers to a set of articles written by colleagues which could be described as in (3).

(3) {x ARTIKEL | $\exists x \in$ MITARBEITER: VERÖFFENTLICHEN (x,y)}

There is no term in representation (2) that has the same denotation as (3). This means that utterances introduce discourse referents which seem not to belong to the semantic representation itself, but rather to the content of the utterance. In the continuation of the dialogue the speaker can refer to these discourse referents, as is in fact done in the last question in (1). We will call these kinds of discourse referents *contextual*.

Because of the occurrence of contextual anaphora in the SPICOS dialogue, an approach to discourse representation has been chosen which, with respect to intersentential anaphora, is very different from DRT (Kamp, 1981). We decided to represent all discourse referents as set terms as part of the context of the utterance. Anaphoric expression in the following utterances can pick up these set terms as their referents. The reference set of the anaphoric expression will then be constructed by using these set terms (for a more detailed discussion cf. Streit (1989); also cf. Section 8.4).

Constructing these sets is not a standard task in logic. Special algorithms are needed to infer representations of discourse referents from the utterance that introduces the discourse referent. These algorithms depend on the structure of the utterance. As an example (see 4), an algorithm is shown that infers a description of a discourse referent which is introduced in a yes-no-question by an indefinite NP. The logical form of the yes-no-question is assumed to contain only existential and universal quantifiers, but no cardinals or other determiners.

(4) We make the following assumptions:
Let $Q_i \in \{\forall, \exists\}$
Let CN_i be a set representing a common noun phrase.
Let p be an n-placed predicate representing a verb
Let $\Psi \equiv Q_1\ x_1 \in CN_1 \dots Q_i\ x_i \in CN_i \dots Q_n\ x_n \in CN_n$,
$p(x_1, \dots, x_i, \dots, x_n)$ the representation of a yes-no-question
For technical reasons we need the following definitions:
Let $\Psi^j\ (x_1, \dots, x_{j-1})$: $\equiv$
$Q_j x_j \in CN_j \dots Q_n\ x_n \in CN_n$
$p\,(x_1, \dots, x_j, \dots, x_n)$ $(1 < j < n+1)$
Let $\Psi^{n+1} \equiv p(x_1, \dots, x_j, \dots, x_n)$
We define recursively a class of formation functions which actually construct the desired representations:
The formation functions
F_i $(1 \leqslant i \leqslant n)$ are defined as follows
if $j < i$ then $F_i\ (\Psi^j)$: $\equiv$
(a) $Q_j x_j \in CN_j F_i(\Psi^{j+1})$ if $Q_j \equiv \exists$
(b) Ψ^j & $\exists x_j \in CN_j F_i(\Psi^{j+1})$ if $Q_j \equiv \forall$
if $j = i$ then $F_i\ (\Psi^i)$: $\equiv \Psi^{j+1}$
Let NP_i be an NP with the reference set CN_i.
If NP_i is indefinite, it introduces the discourse referent D_i as follows:
D_i : $\equiv \{x_i \in CN_i \mid F_i\ (\Psi)\}$

The algorithm produces a more complicated expression, as is given in (3). This accounts for the fact that the representation of the (plural) discourse referent introduced in (1) by the indefinite NP *einen Artikel* should be empty if the question is answered with *no*.

8. Presuppositions

In asking a question, a speaker also makes it clear what kind of answer he is expecting. The form of the answer is fixed by the utterance type of his question. Yes–no-questions are expected to lead either to an affirmation, or to a negation; wh-questions are expected to lead to answers which specify the objects which are addressed by the wh-element.

Within a natural language system the intention is to represent questions by expressions, the evaluation of which leads to the answer in the conventionally expected form.

Proceeding in this way presupposes that the expected answer will always be the conversationally correct one. This is not necessarily so. Questions usually contain presuppositions about the domain of the discourse, and these are not expected to be answered when the questions are put. These presuppositions can be either wrong or controversial. This being the case, the question as formulated either cannot be answered, or the answer leads to misunderstandings. In such cases the normal conventionally expected answers do not represent a conversationally adequate reaction.

In question (1), the speaker wants to know whether it is true that Eibl took part in the meeting referred to, or not.

(1) Hat Eibl an dem Treffen von SPICOS im Januar 89 teilgenommen?
(Did Eibl take part in the SPICOS meeting in January 1989?)

The speaker expects an affirmation or a negation of this question. (2) represents a possible but quite taciturn way of answering. The answer which is produced in the SPICOS system is shown by (3). (3) is just a syntactical transformation of the answer.

(2) Nein, das is nicht der Fall.
(No, that is not the case.)

(3) Nein, Eibl hat nicht an dem Treffen von SPICOS im Januar 89 teilgenommen.
(No, it was not the SPICOS meeting in January 1989 that Eibl took part in.

But the answer is obviously misleading if in January 89 there was no SPICOS meeting. Nothing prevents the speaker from continuing the dialogue with further questions about a non-existent SPICOS meeting.

(4) Hat De Vet an dem Treffen von SPICOS im Januar 89 teilgenommen?
(Did De Vet take part in the SPICOS meeting in January 1989?)

The reason for this misunderstanding is the following: The speaker presupposes in question (1) the existence of a certain SPICOS meeting, a fact that can be inferred from the use of the definite article (*dem* Treffen). Answer (2) gives no hint to the questioner that he is making a presupposition which does not fit. Answer (3) confirms the incorrect presupposition by using the critical NP with the definite article. "Presuppositions are assumed to survive semantically external negation unless there is evidence to the contrary, in which case it is blocked". This is a statement about presupposition in statements (Horton and Hirst, 1988), but it fits the situation perfectly where a yes–no-question is answered by a simple no.

8.1. Presuppositions in the SPICOS dialogue

In the SPICOS dialogue not every phenomenon appears that is possible in connection with the presuppositions. The reasons for this are on the one hand the limited scope of the syntactic constructions that can be effected in SPICOS questions and on the other, the characteristics of a database-query dialogue.

(a) Accommodation for presuppositions
Basically, presuppositions can be used to give the listener information. But it is feasible to start from the assumption that the user will not try to use presupposition as a method to update the database.

(b) The projection problem
The projection problem means that the presup-

positions relating to a constituent of a sentence are not always the presuppositions of the whole sentence. In (5) it is not presupposed that *Wilma beat Fred*, though the first constituent – considered in isolation – does so.

(5) Entweder Wilma hat aufgehört, Fred zu schlagen, oder Wilma hat Fred nie geschlagen.
(Either Wilma has stopped beating Fred or Wilma has never beaten Fred.)

The relevant constructions in which the projection problem appears do not occur in the limited dialogue of the SPICOS system. We can therefore start from the assumption that the projection of presuppositions is trivial in the SPICOS questions.

The presuppositions relevant in SPICOS are propositions concerning the existence and cardinality of database objects which are referred to by NPs. In (6), the uniqueness presupposition is expressed that occurs in (1) with the NP *dem Treffen von SPICOS im Januar 89* (We assume that this definite NP is not anaphoric.)

(6) card ({x ∈ Treffen von SPICOS | von (x,JANUAR 89)} = 1

8.2. Implicatures with indefinite NPs

One of the remarkable features of presupposition is that they cannot usually be defeated by negation. In a dialogue, however, a kind of proposition has to be taken into consideration that seems to be true to the speaker only if his question is confirmed. Such propositions occur with indefinite NPs. We will call them indefiniteness implicatures.

(7) Hat Tropf einen Artikel über Syntax geschrieben?
(Has Tropf written an article on syntax?)

(8) Ja.
(Yes.)

(9) Hat er den Artikel veröffentlicht?
(Has he published the article?)

Suppose that Tropf had in fact written not only one article about syntax, but seven. First we want to point out that *no* cannot be answered to (7). The answer *no* might be justified with question (10), where cardinality is in the focus of the question.

(10) Hat Tropf *nur einen* (oder: *genau einen*) Artikel über Syntax geschrieben?
(Has Tropf written only one (exactly one) article on syntax?)

But even the cooperative answer *no, he has written seven articles* sounds strange in connection with (7). On the other hand one cannot argue that *yes* is an inadequate answer to (7) as it would be in the case of an incorrect presupposition or an incorrect cardinal implicature. In natural dialogue, *yes* is a quite normal answer. This holds good because the affirmative answer in this case only refers to the mere existence of an article by Tropf on the subject of syntax.

Nevertheless, the user usually infers from (8) that it is felicitous to assume that there is just one article, to which he can refer by means of an anaphoric singular NP. (In fact the case is more subtle, since there are different implicatures if an indefinite NP lies within the scope of a quantifier.)

Incorrect indefinitess implicatures lead to incorrect presuppositions concerning the number of anaphoric NPs. To correct an incorrect indefiniteness implicature human listeners sometimes wait till an incorrect presupposition, triggered by the incorrect implicature, occurs in the dialogue. In SPICOS a cooperative answer will be given like (11).

(11) Ja, Tropf hat *Artikel* über Syntax geschrieben.
(Yes, Tropf has written articles on syntax.)

8.3. Presuppositions and anaphora

Presuppositions occur with definite NPs. Definite NPs are often anaphoric. With anaphoric NPs, objects are referred to which are partly described in the preceding dialogue.

(12) Wieviele Vorträge über Syntax hat Tropf gehalten?

(How many lectures on syntax has Tropf given?)

(13) Tropf hat sieben Vorträge über Syntax gehalten.
(Tropf has given seven lectures on syntax).

(14) Hat er den Vortrag über Relativsätze in Eindhoven gehalten?
(Did he give the lecture on relative clauses in Eindhoven?)

We address the set of objects an NP refers to (or which is the source of quantification with respect to this NP) as its *reference set.* The lecture mentioned in question (14) is likely to refer to the lectures mentioned in (12) i.e. the lectures by Tropf on syntax. Thus the reference set of the NP *den Vortrag über Relativsätze (the lecture on relative clauses)* differs from the extension of the common noun phrase *Vortrag über Relativsätze (lecture on relative clauses).* The reference set can be paraphrased by *lectures by Tropf on syntax which are on relative clauses.*

The presupposition of existence included in the definite NP *den Vortrag über Relativsätze* obviously refers to *lectures by Tropf on syntax which are on relative clauses* i.e. the reference set of the NP. The same holds good for the uniqueness presupposition which is included in the NP *den Vortrag über Relativsätze.*

With indefinite NPs the situation is more complicated, even when we assume that indefinite NPs are never anaphoric (this assumption is a vast but customary simplification). With indefinite NPs a distinction has to be drawn between the extension of the common noun phrase of the NP (which is identical with the reference set in the examples) and the *referential aspect* of the NP: In dialogues (15) to (17), the extension of the common noun phrase *lecture on syntax* is every lecture on syntax (as far as it is contained in the database). The referential aspect of the NP *lectures on syntax* can be paraphrased by *lectures on syntax, given by a SPICOS colleague.* Cases of anaphora which have this NP as an antecedent actually refer to the referential aspect, as the reader may infer from the dialogues (15) to (17). In other words: the reference set of a pronoun is the referential aspect of its antecedent.

(15) Hat ein Mitarbeiter von SPICOS Vorträge über Syntax gehalten?
(Has a colleague from SPICOS given lectures on syntax?)

(16) Ja, ein Mitarbeiter von SPICOS hat Vorträge über Syntax gehalten.
(Yes, a colleague from SPICOS has given lectures on syntax.)

(17) Wurden diese Vorträge veröffentlicht?
(Have these lectures been published?)

The answer (16) confirms that the common noun phrase as well as the referential aspect of the NP *Vorträge über Syntax* have non-empty extensions. The same holds good for the NP *ein Mitarbeiter von SPICOS.*

In the case of a negative answer the user makes no assumption that there has been a lecture on syntax given by a colleague from SPICOS or, indeed, that there is a colleague from SPICOS who has given a lecture on syntax. On the other hand, the user does not assume that there is no colleague from SPICOS, even in the case of a negative answer.

This means that presuppositions of existence with indefinites are concerned with the extension of the common noun phrase of the indefinite.

The answer given in the case of a failed presupposition relating to existence will be as in (18).

(18) Es gibt überhaupt kein CN.
(There is no CN at all.)

Looked at more closely, the speaker does not always presuppose existence with indefinite NPs. For example, we cannot infer from the indefinite rhematic NP *lectures on syntax* in (18) that the speaker presupposes anything. On the other hand, the listener makes no mistake when he assumes that there is a presupposition on the part of the speaker.

(19) No colleague from SPICOS has given lectures on syntax.

(20) There is no lecture on syntax at all.

Answer (20) implies answer (19), and is therefore the better answer.

The presuppositions that are handled in the SPICOS system turn out to be propositions about the reference set of the NP which contains the presuppositions.

The reference set of NPs is constructed at some stage of semantic representation. The types of presuppositions (existence, uniqueness, etc.) contained in an NP have to be inferred from the syntactic and semantic features of the NP (e.g. definiteness).

8.4. Cooperative answers and presuppositions in SPICOS

An answer will be called *copperative in the narrow sense* if it does not answer the original question, but a modified version of it. It is important that the user should be able to realize what modification has been made.

(21) Hat Schmidbauer die Berichte über Koartikulation an Hoege geschickt?
(Has Schmidbauer sent the reports on coarticulation to Hoege?)

Suppose that there is only one report of this kind. In this case (22) would be a cooperative answer to (21), that is, an answer to the modified question (23).

(22) Ja, Schmidbauer hat den Bericht über Koartikulation an Hoege geschickt.
(Yes, Schmidbauer has sent the report on coarticulation to Hoege.)

(23) Hat Schmidbauer den Bericht über Koartikulation an Hoege geschickt?
(Has Schmidbauer sent the report on coarticulation to Hoege?)

Incorrect presuppositions that do not allow cooperative modification to the question will be called *fatal* (fatality thus depends in turn on the facility of the system to modify the question).

The presupposition of plurality is not fatal. In example (21) an incorrect presupposition of plurality occurs. (22) shows the cooperative answer to (21).

Questions which include incorrect presuppositions of existence cannot in general be modified successfully. They are therefore regarded as fatal. But fatality does not mean that communication breaks down. If there is no report about coarticulation at all, the system will tell the user this fact. To this extent the system always gives a *cooperative answer in the wide sense.*

Evaluation of the representation of the user's questions will lead to the expected answers. To determine the way of answering in cases of presupposition failures, presuppositions have to be checked before any answer evaluation.

In SPICOS, answering a question is a two-stage process. First, the presupposition has to be evaluated in the database. In the case of a fatal presupposition, the user is informed and the system expects the next question. In the case of a non-fatal presupposition, a modified question is generated which leads to a cooperative answer.

9. Dialogue handler and answer generation

Depending on the result of the syntactic analysis, the semantic representation and the presuppositions of the input question, the dialogue handler decides on the type of dialogue act it is going to carry out. This may be a database query with subsequent information to the user on the result, or it may be some other action like disambiguation or clarification in cases of uncertain input. If a query is answered, the answer pattern is generated according to the type of input question (y/n-question, wh-question, etc.), the result of the database evaluation and the type of presupposition failures contained in the sentence. The answer pattern is produced by an appropriate transformation of the structure of the input question (De Vet et al., 1989). The dialogue handler and the answer generation module have been developed by IPO. For a more detailed description the reader is referred to De Vet and Van Deemter (1988).

10. Conclusion

With SPICOS II, a system has been developed that allows a user to access a database in a cooperative spoken dialogue. Task-oriented systems or systems with a clear situational context, such as train timetable inquiries, allow a more or less system-guided dialogue. In contrast to those, the SPICOS II user is free at each point in the dialogue either to continue with a current topic, or to switch to a different one. The dialogue handler does not guide the user, but only takes care of the ongoing communication either by interactively repairing communication "failures" or by interpreting user presuppositions in a cooperative manner.

This is achieved by representing the user's presuppositions and the semantic content of his utterances in formal logic expressions, and by providing various schemes for cooperative answering. The formal representations are derived from the results of a semantically enriched sentence structure, the resolution of anaphoric references and the formal representation of discourse referents. The cooperative answer is derived from presupposition types and the results of database evaluation.

It has been shown how syntactic and semantic constraints can be applied simultaneously during a chart-based parsing process while maintaining their conceptual independence. In particular, the manner has been outlined in which the syntactic component of SPICOS II interacts with acoustic recognition, and how the relatively free word order in German can be handled in a speech-understanding system using an APSG. Furthermore, it has been demonstrated how incorrect word hypotheses are effectively excluded without losing linguistic generality. It has also been shown how formal semantics can be used to improve the dialogue capabilities of a speech system.

In February 1990 we completed a working demonstration prototype of the system, which is now being tested and improved with a set of 50 example dialogues.

One of the limitations of the current SPICOS II system is certainly that it expects grammatically correct input. In the long run we will have to take into account ungrammatical utterances, breaks, "ah's", and the like. The acoustic uncertainty in recognition will remain and has to be supported by knowledge of the restrictions in natural language. The less certain acoustic recognition is, the more restrictive the linguistic element has to be in what type of language it allows for a given application. This is in contradiction to the expected "free" use of natural language in computer systems. How the chosen subset of natural language will be accepted by the user largely depends on an as yet unknown, but supposedly very subtle, balance of recognition rate, language coverage, error recovery and the strength of the application inherently to limit the mode and style of "artificial communication".

References

H.-U. Block and H. Haugeneder (1986), "The treatment of movement-rules in an LFG-parser", *Proc. Coling '86, Bonn.*

W.J. Bronnenberg, H.C. Bunt, S.P.J. Landsbergen, R.J.H. Scha, W.J. Schoenmakers and E.P.C. van Utteren (1980) "The question answering system PHLIQA1", in *Natural Communication With Computers*, ed. by L. Bolc (Mac Millan, London).

H.C. Bunt, R.J. Beun, F.J.H. Dols, J.A. van der Linden and G.O. thoe Schwaarzenberg (1984) "The TENDUM dialogue system and its theoretical basis", *IPO Annual Progress Report 19.*

H.C. Bunt (1985), *Mass terms and model-theoretic semantics* (Cambridge University Press, Cambridge).

R. Chaffin and D. Herrmann (1984). "The similarity and diversity of semantic relations", *Memory and Cognition*, Vol. 12.

R.E. Frederking and M. Gehrke (1988). "Resolving anaphoric reference in a DRT-based dialogue system", *Proc. 4. Österreichische Artificial-Intelligence-Tagung. Wien, 29–31 Aug. 1988*, ed. by H. Trost (Springer, Berlin).

M. Gehrke (1989), "Begriffsdefinition und Referenzidentifikation definiter Beschreibungen", *Proc. 5. Österreichische Artificial-Intelligence-Tagung, Igls/Tirol, March 1989*, ed. by J. Retti and K. Leidlmair (Springer, Berlin).

D. Horton and G. Hirst (1988) "Presuppositions as beliefs", *Coling Budapest, Proc. of the 12th Int. Conf. on Computational Linguistics*, Vol. 1.

L. Karttunen (1981). "Unbounded dependencies in phrase structure grammar: Slash categories vs. dotted lines", in *Formal Methods in the Study of Language, Vol. 2*, ed. by J.A.G. Groenendijk et al. (Mathematisch Centrum, Amsterdam).

B. Lowerre and R. Reddy (1980), "The HARPY speech understanding system", in *Trends in Speech Recognition*, ed. by W.A. Lea (Prentice-Hall, New York).

D. Mergel and A. Paeseler (1987), "Construction of language models for spoken database queries". *Proc. IEEE ICASSP-87, Dallas*, Vol. 2.

R. Montague (1974), "Formal philosophy, selected papers", *Approaches to Natural Language* (Yale University Press, New Haven, CT).

G.Th. Niedermair (1986), "Divided and valency-oriented parsing in speech understanding", *Proc. Coling '86*, Bonn.

G.Th. Niedermair (1987), "Syntactic analysis in speech understanding", *Proc. Eurospeech 87, Edinburgh*, Vol. 1, ed. by J. Laver and M.A. Jack.

A. Paeseler and H. Ney (1989) "Continuous-speech recognition using a stochastic language model", *Proc. IEEE ICASSP-89, Glasgow*, Vol. 2.

M. Poesio and C. Rullent (1987), "Modified caseframe parsing for speech understanding systems", *Proc. IJCAI 87, Vol. 2*, ed. by J. McDermott (M. Kaufmann, Los Altos, CA).

G. Sagerer and F. Kummert (1988), "Knowledge based system for speech understanding", in *Recent Advances in Speech Understanding and Dialogue Systems*, ed. by H. Niemann, M. Lang and G. Sagerer (Springer, Berlin).

Ch. Schmandt and B. Arons (1986), "A robust parser and dialog generator for a conversational office system", *Proc. AVIOS Cong. 1986.*

M. Shigenag et al. (1986), "A speech recognition system for continuously spoken Japanese sentences – SPEECH YAMANASHI", *Transactions of the IEEE of Japan*, Vol. E 69 No. 5.

V. Steinbiss (1989), "Sentence-hypotheses generation in a continuous-speech recognition system", *Proc. Eurospeech 89, Paris, Vol. 2*, ed. by J.P. Tubach and J.J. Mariani.

M. Streit (1989a), "Presuppositions and anaphora in a question-answering speech system", *Proc. Eurospeech 89, Paris, Vol 2*, ed. by. J.P. Tubach and J.J. Mariani.

M. Streit (1989b) "Diskursrepräsentation im Datenbank-Dialog SPICOS II", *Proc. GWAI-89*, 13th German Workshop on Artificial Intelligence, ed. by D. Metzing (Springer, Berlin).

G. Thurmair (1986), "Linguistische Analyse im Projekt SPICOS", *Kleinheubacher Berichte 29*, ed. by Fernmeldetechnisches Zentralamt (Darmstadt).

G. Thurmair (1988), "Semantic processing in speech understanding" in *Recent Advances in Speech Understanding and Dialogue Systems*, ed. by H. Niemann et al. (Springer, Berlin).

J. de Vet and K. van Deemter (1988), *A Dialogue Handler for SPICOS II*. IPO-Report No. 24 (Eindhoven).

J. de Vet, M.M. de Brouwer and P.A.C. Verkoulen (1989), *The SPICOS-2 Answer Generator*, IPO-Report 703 (Eindhoven).

T. Winograd (1983), *Language as a Cognitive Process, Vol. 1: Syntax* (AddisonWesley, Reading).

M.E. Winston, R. Chaffin and D.J. Herrmann (1987), "A taxonomy of part-whole relations", *Cognitive Science*, Vol 11.

S.J. Young, N.H. Russel and J.H.S. Thornton (1988), "Speech recognition in VODIS II", *Proc. IEEE ICASSP-88, Dallas*.