

## 12. LEXICAL STRUCTURES AND CONCEPTUAL STRUCTURES

Wittgenstein's theory of language games has major implications for both computational linguistics and semantic theory. It suggests that the ambiguities of natural language are not the result of careless speech by uneducated people. Instead, they result from the fundamental nature of language and the way it relates to the world: language consists of a finite number of words that may be used and reused in an unlimited number of language games. The same words may be used in different games to express different kinds of things, events, and situations. To accommodate Wittgenstein's games, this paper draws a distinction between lexical structures and deeper conceptual structures. It suggests that words are associated with a fixed set of lexical patterns that remain the same in various language games. The meanings of those words, however, are deeper conceptual patterns that may vary drastically from one game to another. By means of metaphor and conceptual refinement, the lexical patterns can be modified and adapted to different language games in order to construct a potentially unlimited number of conceptual patterns.

### 1. THE EFFECT OF LANGUAGE GAMES ON MEANING

In the classical view of language, semantic theory requires an ontology of all the concepts (or predicates) expressed by the words of a language. Words have associated lexical information about their parts of speech and their obligatory and optional adjuncts. Concepts are organized in structures that represent knowledge about the world: a hierarchy of concept types; Aristotelian definitions of each type by genus and differentiae; canonical graphs or frames that express the constraints on related concepts; and meaning postulates or rules that express the implications of the concepts. Then the lexicon maps words to concepts, listing multiple concept types for words that have more than one meaning. With many variations of notation and terminology, this view has formed the basis for most systems in computational linguistics:

- From the earliest days of machine translation, theorists have sought a universal system of concepts for the elusive *interlingua*, which would serve as an intermediate language for the translation of any natural language into any other natural language.
- Margaret Masterman's original semantic networks (1961) were designed as an ontology for an interlingua. She constructed a lattice of concept types defined in terms of 100 primitives, which she intended as universal.
- Terry Winograd's SHRDLU (1972) is a famous example of a fixed mapping between word and concept types with a built-in mechanism for defining new types.
- Richard Montague (1974) formulated the purest expression of the classical approach in his system of grammar and logic, which deliberately set out to treat "English as a formal language."
- Roger Schank and his students (1975) were strongly opposed to logic-based approaches like Montague's, but their theory of conceptual dependencies was just as classical. Their MARGIE system used only 11 primitive acts as a basis for defining all conceptual relationships.
- Natural language query systems map a small vocabulary (usually less than 5,000 words) to a fixed set of concept types that represent the entities, attributes, and relationships in a database.

These systems have formed the basis for impressive prototypes. Yet none of them have been general enough to be extended from small prototypes to broad-coverage language processors:

- Winograd's book on SHRDLU was entitled *Understanding Natural Language*, but he has now repudiated that title (Winograd and Flores 1986). He denies that SHRDLU or any other system built along classical lines could truly be said to understand natural language.
- Schank now admits that language understanding is much harder than he had thought. For the past decade, he and his students have used a much larger range of concept types without bothering to give explicit definitions in terms of primitives.
- The most widely used machine translation systems are not based on universal interlinguae. Instead, it has proved easier to implement simpler, but often *ad hoc* transfer schemes between pairs of languages.
- Many computational linguists believe that unrestricted language understanding is impossible or at least impractical with current means. Instead, they have restricted themselves to designing processors for

limited domains (Kittredge and Lehrberger, 1982).

- Harris (1968, 1982) has long maintained that specialized grammars must be written for the various "sublanguages" used in science. He believed that recognition of distinct sublanguages of each natural language is a theoretical necessity, not just a practical expedient.

The limitations of classical systems could be attributed either to fundamental flaws in the approach or to temporary setbacks that will eventually be overcome. Some computational linguists, especially the logicians who follow Montague, are still pursuing the classical ideal with newer theories, faster computers, and larger dictionaries. Others who once believed that language was more tractable eventually lost faith and became some of the most vocal critics. Bar-Hillel (1960) was one of the early apostates, and Winograd is one of the most recent.

The most famous apostate who abandoned the classical approach was Ludwig Wittgenstein. His early philosophy, as presented in the *Tractatus Logico-Philosophicus*, was an extreme statement of the classical view. It started with the sentence "The world is everything that is the case" – i.e. a collection of atomic facts about relationships between elementary objects. Atomic facts could be combined to form a compound proposition, which was "a function of the expressions contained in it." Language for him was "the totality of all propositions." He regarded any statement that could not be built up in this way as meaningless, a view that culminated in the final sentence of the *Tractatus*: "Whereof one cannot speak, thereof one must be silent." Wittgenstein's early philosophy was the inspiration for Tarski's model-theoretic semantics, which Tarski's student Montague applied to natural language.

In his later philosophy, as presented in the *Philosophical Investigations*, Wittgenstein repudiated the "grave mistakes in what I wrote in that first book." He completely rejected the notion that all of language could be built up in a systematic way from elementary propositions. Instead, he presented the view of language as a "game" where the meaning of a word is determined by its use. If there were only one set of rules for the game, a modified version of the classical approach could still be adapted to it. But Wittgenstein emphasized that language is not a single unified game, but a collection of as many different games as one can imagine possible uses. "There are *countless* kinds: countless different kinds of use of what we call "symbols", "words", "sentences". And this multiplicity is not something fixed, given once and for all; but new types of language, new language games, as we may say, come into existence, and others become obsolete and get forgotten." As examples of the multiplicity of language games, he cited "Giving orders, and obeying

them; describing the appearance of an object, or giving its measurements; constructing an object from a description (a drawing); reporting an event; speculating about an event; forming and testing a hypothesis; presenting the results of an experiment in tables and diagrams; making up a story, and reading it; play acting; singing catches; guessing riddles; making a joke, telling it; solving a problem in practical arithmetic; translating from one language into another; asking, thanking, cursing, greeting, praying." He regarded this view as a complete rejection of "what logicians have said about the structure of language," among whom he included himself.

Wittgenstein's language games were the inspiration for speech act theory, which has become one of the major topics in pragmatics. Their implications for semantics, however, are just as important. As an example, consider the verb *support* in the following sentences:

Tom supported the tomato plant with a stick.  
 Tom supported his daughter with \$8,000 per year.  
 Tom supported his father with a decisive argument.  
 Tom supported his partner with a bid of 3 spades.

These sentences all use the verb *support* in the same lexical pattern:

A person supported NP<sub>1</sub> with NP<sub>2</sub>.

Yet each use of the verb can only be understood with respect to a particular subject matter or domain of discourse: physical structures, financial arrangements, intellectual debate, or the game of bridge. Each domain has its own language game, but they all share a common vocabulary and syntax. The meanings of the words, however, change drastically from one domain to the next. As a result, the mapping from language to reality is indirect: instead of the fixed mappings of Montague grammar, the mapping from word to reality may vary with every language game.

Both Wittgenstein's philosophical analyses and thirty years of experience in computational linguistics suggest the same conclusion: a unified semantic basis along classical lines is not possible for any natural language. Instead of assigning a single meaning or even a fixed set of meanings to each word, a theory of semantics must permit an open-ended number of meanings for each word. Following is a sketch of such a theory:

- Words are like playing pieces that may be used and reused in different language games.
- Associated with each word is a limited number of lexical patterns that determine the rules that are common to all the language games that use the word.

- Meanings are deeper conceptual patterns that change from one language game to another.
- Metaphor and conceptual refinement are techniques for transferring the lexical patterns of a word to a new language game and thereby creating new conceptual patterns for that game.

As an analogy, consider the Japanese games of go and go-moku, both of which use the same board, the same pieces, and the same superficial playing patterns: the board is lined with a 19 by 19 grid; the pieces consist of black stones and white stones; and starting with an empty board, two players take turns in placing stones on the intersections of the grid. At this purely syntactic level, the two games are the same. At a semantic level, however, there are profound differences in the meanings of the patterns of stones: in go, the goal is to form "armies" of stones that surround territory; in go-moku, the goal is to form lines with five consecutive stones of the same color. Although the same moves are syntactically permissible in the two games, the semantic differences cause very different patterns to emerge during play. In the analogy with language, the stones correspond to words, and the two games correspond to different domains of discourse that happen to use the same words. At a syntactic level, two different games may permit words or pieces to be used in similar ways; but differences in the interpretation lead to different meanings for the combinations. To continue the analogy, new games may be invented that use the same pieces and moves. In another game, the player with the black stones might try to form a continuous path that connects the left and right sides of the board, while the player with white would try to connect the top and bottom. The syntax would be the same as in go and go-moku, but the meanings of the patterns of stones would be different. Just as old pieces and moves can be used in new games, language allows old words and syntax to be adapted to new subjects and ways of thinking.

## 2. INTERACTIONS OF THE LEXICAL AND CONCEPTUAL SYSTEMS

Each natural language has a well-organized lexical and syntactic system. Each domain of knowledge has a well-organized conceptual system. Complexities arise because each language tends to use and reuse the same words and lexical patterns in many different conceptual domains. In his discussion of sublanguages, Harris (1968) cited the following two sentences from the domain of biochemistry:

The polypeptides were washed in hydrochloric acid.  
 \*Hydrochloric acid was washed in polypeptides.

Harris observed that both of them could be considered grammatical as examples of general English sentences. But he claimed that grammatical restrictions in the sublanguage of biochemistry permitted the first one and excluded the second. Harris's observations about permissible sentences in biochemistry are correct, but he attributed too much to grammar. What makes the second sentence unacceptable are facts about chemistry, not about grammar. As in the games of go and go-moku, the syntax permits either combination, but knowledge of the subject matter determines which patterns are likely or unlikely.

In Harris's sentences, the syntax clearly indicates what is washed and what is being washed. Noun-noun modifiers, however, provide no syntactic clues, and domain knowledge is essential for understanding them. The following two noun phrases, for example, both use the noun *wash* in the sense of a liquid used to wash something:

a hydrochloric acid wash  
 a polypeptide wash

The surface syntax is the same in both. Only knowledge of the domain leads to the expectation that hydrochloric acid would be a component of the liquid and polypeptides would be washed by the liquid. A Russian or Chinese chemist with only a rudimentary knowledge of English could interpret these phrases correctly, but an English-speaking linguist with no knowledge of chemistry could not.

Harris's example illustrates the interactions of the lexical and conceptual systems. An English-speaking chemist and an English-speaking linguist would share common lexical and syntactic habits, but the conceptual patterns for their specialties would be totally unrelated. An American, Russian, and Chinese chemist, however, would have no shared lexical and syntactic patterns, but their conceptual patterns in the field of chemistry would be similar. For technical terms like *hydrochloric acid* or *polypeptides*, which are used only in a narrow domain, an MT system can easily provide an accurate translation. More difficult problems occur with common words that are used in many different domains in slightly different ways. One Russian-to-English MT system, for example, gave the translation *nuclear waterfall* for what English-speaking physicists call a nuclear cascade. A specialized technical word like *nuclear* has a unique translation, but a more common word like *waterfall* or *cascade* has more uses in more domains and consequently more possible translations.

The main reason why the correct word sense is hard to determine is that different senses often occur in the same syntactic and lexical patterns. The

examples with the verb *support* all used exactly the same pattern. Yet Tom performed totally different actions: using a stick to prop up the tomato plant; giving money to his daughter; and saying something that made his father's statements seem more convincing. Physical support is the basic sense of the word, and the other senses are derived by metaphorical extensions. In other languages, the basic vocabulary may have been extended by different metaphors. Consequently, different senses that all use the same pattern in English might be expressed with very different patterns in another language. Russian, for example, would use the following constructions:

Tom placed a stick in the ground in  
 order to support  
 [podd'eržat'] the tomato plant.  
 Tom spent \$8,000 per year on the support  
 [sod'eržanie] of his daughter.  
 Tom supported [podd'eržal] his father with  
 [instrumental case] a decisive argument.

Russian uses the verb *podd'eržat'* in different syntactic constructions for the first and third sentences. For the second, it uses a noun *sod'eržanie* derived from a related verb *sod'eržat'*. As these sentences illustrate, different uses of a word may be expressed with the same lexical and syntactic patterns in one language, but the translations into another language may use different words in very different patterns.

The translation from English to Russian also illustrates another point: human translators often add background knowledge that is implicit in the domain, but not stated in the original words. For this example, the Russian lexical patterns required an extra verb in two of the sentences. Therefore, the translator added the phrase *placed a stick in the ground* in the first sentence and the verb *spent* in the second. The verbs *place* and *spend* and the noun *ground* did not occur in the original, but the translator felt that they were needed to make natural-sounding Russian sentences. A syntax-based MT system could not add such information, which can only come from background knowledge about the domain. (The term *commonsense* is often used for background knowledge, but that term can be misleading for detailed knowledge in technical domains – most people do not have any commonsense intuitions about polypeptides.)

As another example, Cruse (1986) cited the word *topless*, as used in the phrases *topless dress*, *topless dancer*, and *topless bar*. Literally, something is topless if it has no top. That definition is sufficient for understanding the phrase *topless dress*. For the other phrases, a young child or a computer system without domain-dependent knowledge might assume that a topless dancer or a topless

bar are somehow missing their own tops. An adult with knowledge of contemporary culture, however, would know that the missing top is part of the clothing of the dancer or of certain people in the bar. Cruse gave further examples, such as *topless by-laws* or *topless watchdog committee*, which require knowledge of even more remote relationships, including public attitudes towards topless behavior. These examples show that domain-dependent knowledge is often essential for determining the relationship between an adjective and the noun it modifies. Computer systems and semantic theories that map adjectives into simple predicates can represent the literal use in *topless dress*, but they cannot interpret any of the other phrases.

For the different uses of *support* and *topless*, the lexical and syntactic patterns are the same, but the conceptual patterns are different. These examples illustrates a fundamental principle: the same lexical patterns are used across many different conceptual domains. The lexical structures are

- Relatively domain independent,
- Dependent on syntax and word forms,
- Highly language dependent.

And the conceptual structures are

- Highly domain dependent,
- Independent of syntax and word forms,
- Language independent, but possibly culture dependent.

When there are cross-linguistic similarities in lexical patterns, they usually result from underlying conceptual similarities. The English verb *give*, for example, takes a subject, object, and indirect object. Other languages may have different cases marked by different prepositions, postpositions, inflections, and word order; but the verbs that mean roughly the same as *give* also have three participants – a giver, a thing given, and a recipient. In all languages, the three participants in the conceptual pattern lead to three arguments in the lexical patterns.

The view that lexical patterns are reflections or projections of underlying conceptual patterns is a widely held assumption in cognitive science: the first lexical patterns a child learns are derived from conceptual patterns for concrete things and events. Actions with an active agent doing something to a passive entity lead to the basic patterns for transitive verbs. Concepts like SAY or KNOW that take embedded propositions lead to patterns for verbs with sentence complements. Once a lexical pattern is established for a concrete domain, it can be transferred by metaphor to create similar patterns in more abstract domains. By this process, an initial set of lexical patterns can be built



up; later, they can be generalized and extended to form new conceptual patterns for more abstract subjects. The possibility of transferring patterns from one domain to another increases flexibility, but it leads to an inevitable increase in ambiguity. If the world were simpler, less varied, and less changeable, natural languages might be unambiguous. But because of the complexity, the meanings of words shift subtly from one domain to the next. If a word is used in widely different domains, its multiple meanings may have little or nothing in common.

### 3. REPRESENTING LEXICAL AND CONCEPTUAL STRUCTURES

Up to this point, the words *meaning*, *pattern*, and *word sense* have been used as informal terms with their commonly accepted English meanings. As with any terms, precision is only possible within a particular conceptual domain – in this case, a formal theory of semantics. For the purpose of this paper, the theory of conceptual graphs (Sowa, 1984) will be used. Conceptual graphs form a complete system of logic; they support inheritance in the same way as frames; they support nested contexts that are equivalent to discourse representation structures (Kamp 1981a,b); and they explicitly show the case relations or thematic roles. The ideas in this paper could be adapted to other forms of logic, such as the predicate calculus, but conceptual graphs show the underlying relationships more clearly.

Lexical structures and conceptual structures could both be represented by conceptual graphs, but they differ in the type labels on the concept nodes of the graphs. For lexical patterns, type labels are taken from surface word forms: the word *support*, for example, could be represented by the concept type SUPPORT. More specialized word senses could be represented by subtypes of SUPPORT, such as SUPPORT-PHYS for physical support or SUPPORT-FIN for financial support. Following are the correspondences between the informal terms and the terms of conceptual graph theory:

- A *lexical type* is a concept type that corresponds to a word form in a natural language, such as SUPPORT for the word *support*. Except for homonyms, which will be discussed in the next section, each content word – noun, verb, adjective, or adverb – has its own lexical type. Prepositions and conjunctions are represented by conceptual relations; quantifiers and other determiners are represented by symbols in the referent field of a concept node.

- Specialized senses such as financial support or physical support are represented by subtypes of a lexical type, such as SUPPORT-FIN as a subtype of SUPPORT. Concept types other than lexical types do not correspond to single words; they would have to be expressed by a word that corresponds to one of their supertypes. Their type labels would normally be hyphenated as in MOBILE-ENTITY or SUPPORT-FIN.
- Not all word senses are represented by distinct concept types. For Cruse's example of different uses for the word *topless*, he claimed that each example used a distinct sense. In conceptual graphs, however, *topless* would be represented by a single lexical type TOPLESS. The different senses correspond to different paths of concepts and relations that link the concept [TOPLESS] to the concepts [DRESS], [DANCER], or [BAR].
- The term *lexical pattern* is an informal term used to represent the kinds of patterns in which a word typically occurs. Those patterns may be represented by conceptual graphs called *canonical graphs*, which show the types of relations that may be attached to a lexical type and the restrictions on the concept types that may be linked to those relations. Canonical graphs correspond to case frames or subcategorization frames for verbs, but they are generalized to all parts of speech.
- The term *conceptual pattern* is another informal term. All conceptual patterns may be represented by conceptual graphs or by parametrized conceptual graphs, as in type definitions and schemata. A single conceptual graph may contain both lexical types and more specialized types.
- The word *meaning* is a catchall term for almost anything associated with a word. In a narrow sense, it corresponds to a type definition for a concept. In a broad sense, it could include emotional connotations as well as any background information commonly associated with a word. In this paper, the term *meaning* is used only in informal discussions, never as a technical term.

As an example, Figure 1 shows a canonical graph that represents the lexical pattern associated with the verb *support* with its lexical type SUPPORT. It shows that every instance of SUPPORT has four expected participants: an animate agent, some entity as patient, some entity as instrument, and a purpose, which is represented by a nested context. That context, which might represent something at a different time and place from the outer context, shows the entity

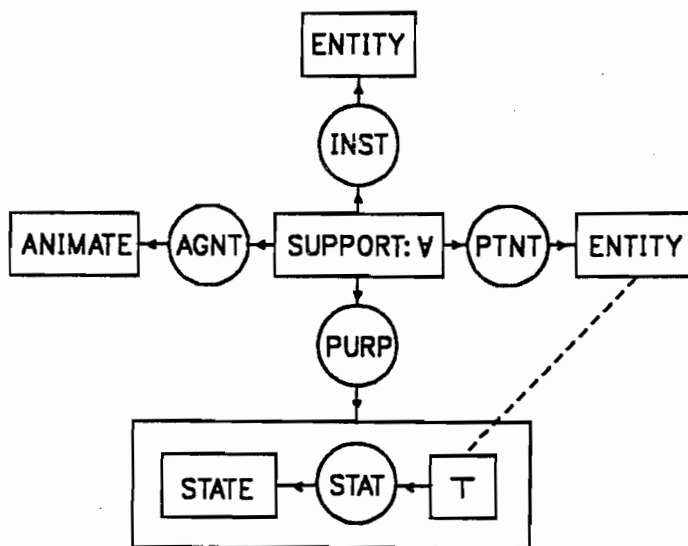


Fig. 1. Canonical graph for the lexical type SUPPORT.

in some state.

Canonical graphs like Figure 1 can express more detailed structural information than the case frames used to show selectional constraints. Whereas the usual case frames merely show the thematic roles for a verb and the expected types that can fill those roles, the graphs can grow arbitrarily large: they can show long-range dependencies far removed from the central concept; and they may contain nested contexts that show situations at different times and in different modalities. The dotted line in Figure 1 is a *coreference link* that crosses context boundaries; it shows that the entity that is the patient of SUPPORT is coreferent with the thing in the nested context. As an example of a more complex graph, Figure 9 in Section 7 has 14 concept nodes, 12 conceptual relations, 3 nested contexts, and 2 coreference links. Figure 1 is already beyond the capabilities of most frame systems; no frame system ever designed can represent the equivalent of Figure 9.

Figure 1 shows the *display form* for conceptual graphs; for convenience in typing, the graph could also be represented in an equivalent *linear form*:

```
[SUPPORT: v] -
(AGNT) → [ANIMATE]
(PTNT) → [ENTITY: *x]
```

(INST) → [ENTITY]  
 (PURP) → [ [\*x] → (STAT) → [STATE]].

In this form, the square brackets represent the concept boxes, and the variable \*x shows the coreference link. Besides the canonical graph, the graph for *support* can also contain *syntactic annotations* to indicate which relations are optional or obligatory (Fargues et al. 1986; Sowa 1991). The next graph, for example, uses a semicolon to divide the semantic information from the syntactic annotations, which indicate that the agent and patient are obligatory, but the instrument and purpose are optional:

[SUPPORT: v]-  
 (AGNT) → [ANIMATE; oblig]  
 (PTNT) → [ENTITY: \*x; oblig]  
 (INST) → [ENTITY; opt] (PURP) → [ [\*x] →  
 (STAT) → [STATE]; opt].

The annotations after the semicolons are not part of the propositional content of a conceptual graph. When conceptual graphs are used as a system of logic, all of the information from the semicolon up to the closing bracket may be erased or ignored.

The canonical graph in Figure 1 shows what is common to every use of the lexical type SUPPORT, but it is too general to determine which subtype is intended in any particular use of the word *support*. Each conceptual domain would require more specialized subtypes of SUPPORT with canonical graphs that would impose tighter constraints. For the game of bridge, the subtype SUPPORT-BRIDGE is highly restrictive: you support your partner, but you overcall your opponents; the supporter and the supportee must stand in a relationship that is highly specific to the domain of bridge. For the example of hydrochloric acid and polypeptides, the constraints are taken from another domain that is just as restrictive. Every time someone invents a new game or makes a new scientific discovery, a new conceptual domain is created or an old domain is refined and enlarged. Such modifications to a conceptual domain change the selectional constraints in English and every other natural language. Since those constraints are so detailed and so domain dependent, they do not belong in a general lexicon of English. Instead, the general lexicon should contain simple lexical patterns like Figure 1, and the more detailed constraints should be kept in separate knowledge bases for each domain.

The lexical type SUPPORT is derived from the English word *support*. The more specialized concept types SUPPORT-FIN and SUPPORT-PHYS are intended to be language independent. SUPPORT-FIN would also be a subtype of SODERZHAT, which is a lexical type derived from the Russian verb *sod'eržat'*.

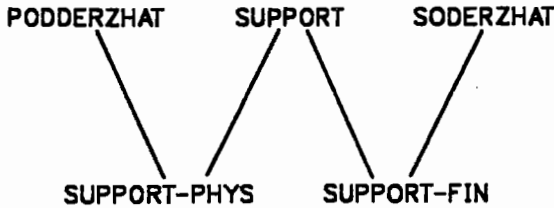


Fig. 2. Type hierarchy for SUPPORT and related concepts.

In a different syntactic pattern, the concept type SODERZHAT could also be expressed as the Russian noun *sod'eržanie*. Similarly, SUPPORT-PHYS would be a subtype of both the English lexical type SUPPORT and the Russian lexical type PODDERZHAT. The next diagram shows an excerpt from the type hierarchy that includes these concept types (Figure 2).

The rules of English syntax would map SUPPORT to and from the English verb or noun *support*. Russian rules would map the types SODERZHAT or PODDERZHAT to or from the corresponding Russian verbs and nouns. Domain-specific rules for the financial domain or the physical domain would permit language-independent inferences from SUPPORT-FIN or SUPPORT-PHYS.

Whereas different senses of *support* are represented by different concept types, the word *topless* requires only a single type for all the senses discussed by Cruse. The lexical type TOPLESS does not require any specialized concept types to express the different meanings. Its canonical graph is quite simple:

$$[\text{TOPLESS}] \leftarrow (\text{ATTR}) \leftarrow [\text{ENTITY}] .$$

This graph shows that TOPLESS is normally linked to an ENTITY by the ATTR (attribute) relation. Following is a type definition that defines TOPLESS as a property of an entity that does not have a top as part:

**type.** TOPLESS (x) is.

$$[\text{PROPERTY: } *x] \leftarrow (\text{ATTR}) \leftarrow [\text{ENTITY: } *y] \neg \\ [[*y] \rightarrow (\text{PART}) \rightarrow [\text{TOP}]] .$$

The variable \*x marks the formal parameter, and the variable \*y shows a coreference link between the concept [ENTITY] and the coreferent concept inside the negation. For the phrase *topless dress*, the canonical graph and the type definition are sufficient to determine the correct interpretation. Unfortunately, these graphs would lead to incorrect interpretations of the other examples *topless dancer* and *topless bar*. To interpret those phrases, start with the following

graph, which says that an article of clothing is worn by a person:

$$\begin{array}{l} [\text{CLOTHING-ARTICLE}] \leftarrow (\text{PTNT}) \leftarrow \\ \quad [\text{WEAR}] \rightarrow (\text{AGNT}) \rightarrow [\text{PERSON}] . \end{array}$$

This graph uses the type CLOTHING-ARTICLE, which is a supertype of DRESS. The lexical type CLOTHING represents a collection of articles of clothing, which may include a dress as part, not as a subtype. Since CLOTHING-ARTICLE is a subtype of ENTITY, this graph may be joined to the canonical graph for TOPLESS to form the next one:

$$\begin{array}{l} [\text{TOPLESS}] \leftarrow (\text{ATTR}) \leftarrow [\text{CLOTHING-ARTICLE}] \leftarrow (\text{PTNT}) \\ \quad \leftarrow [\text{WEAR}] \rightarrow (\text{AGNT}) \rightarrow [\text{PERSON}] . \end{array}$$

This graph says that topless is an attribute of a clothing article worn by a person. It could be used to interpret both phrases *topless dress* as well as *topless dancer*. Since a dancer is a type of person, not a clothing article, the concept [PERSON] in this graph would be restricted to [DANCER] to represent the topless dancer. The phrase *topless bar* would require another graph for the concept type BAR:

$$\begin{array}{l} [\text{BAR}] \leftarrow (\text{LOC}) \leftarrow [ [\text{PATRON: } \{*\} *x] \leftarrow (\text{AGNT}) \leftarrow [\text{DRINK}] \\ \quad [ *x ] \leftarrow (\text{RCPT}) \leftarrow [\text{SERVE}] \rightarrow (\text{AGNT}) \rightarrow [\text{PERSON: } \{*\} ] \\ \quad [ *x ] \leftarrow (\text{BENF}) \leftarrow [\text{ENTERTAIN}] \rightarrow (\text{AGNT}) \rightarrow [\text{PERSON: } \{*\} ] ] . \end{array}$$

This graph says that a bar is a location of a situation represented by the nested context. In that context, a set \*x of patrons drink (the symbol {\*} represents a set of unspecified individuals), the same \*x's are the recipients of service by another set of persons, and they are also the beneficiaries of entertainment by some other set of persons. This graph shows three kinds of people who could be wearing topless clothing: the patrons, the servers, and the entertainers. The phrase *topless bar* could be interpreted by joining the concept [PERSON] in the previous graph to any one of the concepts representing people in this graph. Cruse's more remote examples of *topless by-laws* or *topless watchdog committee* could also be interpreted by a similar process of joining further graphs to the original canonical graph for TOPLESS.

These examples show how multiple word senses can arise when a single concept type is used in different ways. The literal sense of the adjective *topless* is represented by the ATTR relation linking the concept [TOPLESS] directly to the concept corresponding to the noun modified by the adjective. The more remote senses correspond to longer and longer paths between [TOPLESS] and the concept corresponding to the noun. Lexicographers use the term *lexeme* for a word used in a particular sense. That term is useful for classifying word senses, but it does not explain them or represent them precisely. A formalism such as conceptual graphs not only shows how the various senses are related to

a common lexical type, but it also provides a basis for computational systems that can work with those types.

The semantic interpreter designed by Sowa and Way (1986) could interpret these examples correctly only if it had a suitable set of graphs. If it started with the simple canonical graph, it would assume that the dress, the dancer, the bar, the by-laws, and the committee were all missing their own tops. But if it started with the graph showing that *topless* was an attribute of some clothing, then it would correctly interpret *topless dancer* as well as *topless dress*. The system is also capable of joining the graph for BAR to the previous graph to interpret the phrase *topless bar*. It could even discover the ambiguity concerning which people were topless. Before it can do that, however, it must start with the right graph. It could also start with a small number of candidate graphs if it had criteria for selecting the best interpretation. An example of a good criterion is that the semantic distance between DANCER and PERSON is much less than the distance between DANCER and ENTITY. Various implementations of conceptual graph systems, including the ones by Fargues et al. (1986) and Garner and Tsui (1988) have implemented such criteria for determining the best match. Levinson and Ellis (1992) have developed efficient methods for searching large knowledge bases with many thousands of graphs.

Cruse's more remote examples, *topless by-laws* and *topless watchdog committee*, would be harder to process by computer since they would require several graphs to be joined. Joining the graphs is not difficult; the hard part is finding which graphs to join. Neither the word *topless* nor the word *by-law* mentions clothing, people, moral standards, public attitudes, or legal issues. Yet all of those concepts would be involved in the graphs needed to interpret the phrases. When people lack the background knowledge, they cannot understand such phrases any more easily than a computer: a person from a culture where topless behavior is normal would require a lot of explanation.

Domain-dependent knowledge is essential to language understanding. People use it constantly; ways of representing it, finding it, and using it must be central to any semantic theory. Furthermore, the aspects of knowledge that are stated explicitly or left implicit may differ from one language to another. Figure 3 shows a translation of a text from language A to language B. The text in A is mapped into graphs with lexical types derived from the words in A. With the help of domain-dependent knowledge, the lexical structures are expanded into deeper conceptual structures, which can then be re-expressed in the lexical structures of language B. Finally, the graphs with lexical types from language B are mapped into a text in B.

Because of their different lexical patterns, different languages treat different

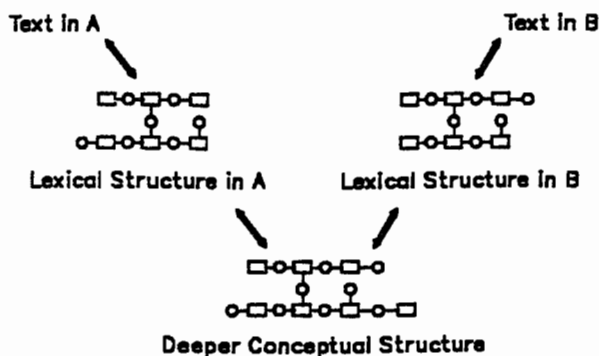


Fig. 3. Translating a text from language A into language B.

kinds of information as optional or obligatory. As a result, the information expressed in language B might not be identical to the information in A. The translation of *support* into Russian, for example, added information about spending money or placing a stick in the ground. In translating English into Chinese, the sentence *Cats like fish* would become *Māo ài chī yú* [cat like eat fish]. In Chinese, plurals are optional, but liking to eat must be distinguished from liking as a friend. English speakers and Chinese speakers have similar knowledge about how cats behave towards fish, but their languages cause them to emphasize different aspects of that knowledge. Such knowledge cannot be introduced syntactically; it must come from background knowledge about the subject matter.

#### 4. THE NATURE OF LEXICAL AMBIGUITY

All natural languages have ambiguities, both syntactic and semantic. The syntactic ambiguities affect the shape of parse trees and are therefore called *structural ambiguities*. There are four major kinds:

- Multiple parts of speech for a single word;
- Different parse trees for the same sentence;
- Unresolved referents for pronouns and definite noun phrases;
- Unclear scopes of quantifiers and negations.

Semantic ambiguities, also called *lexical ambiguities* since they depend on the meanings of words, have been largely neglected in formal theories. Their



origin and nature, however, touch upon a number of central issues that must be addressed by any theory of semantics. There are two major kinds of lexical ambiguities:

- Homonymy, where two or more historically distinct words happen to acquire the same pronunciation and often the same spelling as well;
- Polysemy, where a word has a number of closely related meanings.

Examples of homonymy include *page* in a book vs. *page* as an attendant or *ball* as a rounded object vs. *ball* as a dance. Polysemy is a more common kind of lexical ambiguity where the differences between senses tend to be small, subtle, and hard to distinguish. One example of polysemy is the word *support* with its multiple meanings that were discussed earlier. Another example is the verb *yield* in the following sentences: *Two molecules of H<sub>2</sub> and one molecule of O<sub>2</sub> yield two molecules of H<sub>2</sub>O. Vehicles approaching from the entrance ramp must yield to oncoming traffic.*

What distinguishes homonymy from polysemy is a clear break in the range of meanings. For polysemous words, different dictionaries usually list different numbers of meanings, with each meaning blurring into the next. For homonyms, however, dictionaries usually agree upon the number of distinct groups of meanings. The word *ball* as a rounded object, for example, is derived from an Old English word with similar meaning; the word *ball* as a dance was borrowed from French in the 17th century. The *page* in a book comes from the Latin *pagina*, and the *page* as an attendant comes from the Italian *paggio*. Each of these homonyms has polysemous variants, but there are no intermediate meanings of *ball* or *page* that blur the distinction between the homonyms. As these examples illustrate, homonyms arise from distinct word forms that accidentally come together, either because of borrowing (as with *ball*) or because of sound changes that lose distinctive features (as with the merger of *pagina* and *paggio* to form *page*).

Unlike homonyms, which result from linguistic processes of borrowing and sound change, polysemous variants result from the complexities of mapping language to the world. As an example of polysemy, consider the term *oil well*. Most dictionaries give only one meaning for the term, and most MT systems would have no difficulty in translating it to another language. Yet one oil company found a serious ambiguity in its definition. In their geological database, an oil well was defined as any hole in the ground drilled or dug for the purpose of obtaining oil, whether or not the hole proved to be dry. In their financial database, however, an oil well was defined as a pipe connected to one or more holes in the ground that produce oil. The financial database therefore



An Italian reader who happened to see this sentence in a newspaper would interpret it in only one way: "The association of the industrialists has approved a new plan of investments in the south of Italy." Yet the sentence is ambiguous at several different points. The first two ambiguities involve words that have more than one part of speech:

- The noun *industriali* could also be an adjective or an imperative form of the verb *industrialiare* with an enclitic pronoun *-li*.
- The noun *piano* could also be an adjective or an adverb.

For this sentence, syntactic rules immediately resolve these ambiguities; they only permit *industriali* and *piano* to be nouns. The sentence also has one structural ambiguity involving modifier attachment:

- The prepositional phrase *nel mezzogiorno* could modify the verb *ha approvato*, the noun *piano*, or the noun *investimenti*.

For this ambiguity, the correct interpretation follows from the heuristic principle of local attachment: the prepositional phrase modifies the closest word in the parse tree that satisfies the selectional constraints, in this case *investimenti*. Finally, the sentence contains three lexical ambiguities:

- The noun *piano* could mean *plane*, *plan*, *project*, *storey* [of a building], or *pianoforte*.
- The noun *investimento* could mean *financial investment* or *traffic accident*.
- The noun *mezzogiorno* could mean *noon*, *south*, or *southern Italy*.

Selectional constraints can only resolve the meaning of the last word: *nel* requires a place, such as southern Italy; the time, at noon, would have to be expressed *à mezzogiorno*. To resolve the other ambiguities, DANTE ignores meanings that rarely occur in finance: it assumes that every occurrence of the word *investimento* means *financial investment*; and it uses a single concept type PROGETTO to subsume the two similar senses of plan and project while ignoring the senses of plane, storey, and pianoforte, which are much less common in the financial domain.

The etymologies of these words illustrate the processes through which words acquire multiple senses. The senses recorded in dictionaries happen to be fossilized records of the major distinctions that were made over the centuries. But the same processes are constantly operating in all uses of language. The noun *piano*, for example, is derived from the Latin *planum* meaning plane surface; that is still one of its central meanings. The meaning of storey is a specialization of plane. The meaning of pianoforte should be considered a separate homonym, since it was derived by abbreviating a distinct word.

The meanings of *plan* and *project* were derived from *plane* by the process of *metonymy*, which involves referring to a thing by something else that is closely associated with it. In this case, a plan is something written on the plane surface of a sheet of paper. The multiple senses of *mezzogiorno* and *investimento* were also derived by metonymy. Literally, *mezzogiorno* means midday or noon. By metonymy, it came to mean south, which is the direction of the sun at noon. That meaning could be represented by a type definition in conceptual graphs:

**type**    MEZZOGIORNO-DIRECTION (\*x) is  
             [DIRECTION: \*x] → (TOWARD) → [SUN] →  
             (TIM) → [MEZZOGIORNO] .

By a second step of metonymy, it could mean the place located in the direction of the sun at noon:

**type**    MEZZOGIORNO-PLACE (\*x) is  
             [PLACE: \*x] → (LOC) → [MEZZOGIORNO-DIRECTION] .

The noun *investimento* is derived from the verb *investire*, which literally means to put on clothing. By metonymy, it came to mean the act of installing a person in office, symbolized by putting on special robes or insignia. By another step of metonymy, the meaning of financial investment arose as an act performed by a person who had already been invested in the office of banker. By one more step, it also came to mean the amount of money involved in an act of investment. Since clothing surrounds a person, *investire* also acquired the meaning of surround. Since an army invests a fortress for the purpose of knocking it down, the verb came to mean knock down or collide with, from which *investimento* came to mean traffic accident. Each step in the derivation is a typical polysemous shift in meaning; but over the centuries, the meanings have diverged so far that they should be treated as clearly distinct concept types, as distinct as any homonyms.

## 5. METONYMY, METAPHOR, AND CONCEPTUAL REFINEMENT

Metonymy and metaphor both violate expected constraints in the literal use of language. The difference between them is that metonymy omits intervening relationships between concept types, but it doesn't change the types themselves. Metaphor, however, extends the type hierarchy and transfers old lexical patterns to the new concept types. The following passage from a science text for children illustrates a typical kind of metonymy:

There are four stages in the life of a butterfly.  
 The first stage is an egg no bigger than  
 the head of a pin.

Most dictionaries say that *stage* could mean platform or time period, but not egg. The meaning of time period was originally derived as a metonym for the time spent waiting at a stage considered as a location; that is an old meaning codified in dictionaries. By a further step of metonymy, *stage* could refer to something associated with a particular time period. In this example, it is a metonym that refers to the body of a butterfly in the first time period of its life.

Proper names are frequently used as metonyms. During a debate, one senator told another *You're no Jack Kennedy*. Since the quantifier *no* cannot normally be used with proper names, it signals a metonym for a person of the same type as Jack Kennedy:

$$(\lambda x) \text{ [PERSON: *x]} \rightarrow (\text{KIND}) \rightarrow \text{[TYPE]} \leftarrow (\text{KIND}) \leftarrow \text{[PERSON: Jack Kennedy]} .$$

This  $\lambda$ -expression defines a kind of person  $x$  who is of the same type as the person Jack Kennedy. Names of buildings or places can also be used as metonyms, as in the sentence *The White House announced the new budget*. This metonym is signaled by a violation of selectional constraints: a building cannot announce anything; instead, some person associated with the White House must have made the announcement. Following is a noncanonical graph that a semantic interpreter might attempt to generate for the phrase *The White House announced...*

$$\text{[ANNOUNCE]} \rightarrow (\text{AGNT}) \rightarrow \text{[BUILDING: White House]} .$$

This graph is inconsistent with the canonical graph for ANNOUNCE, which would require a person as the agent. To resolve the conflict, the interpreter could insert a concept [PERSON] linked to [ANNOUNCE] by AGNT and to [BUILDING] by an unknown relation R:

$$\text{[ANNOUNCE]} \rightarrow (\text{AGNT}) \rightarrow \text{[PERSON]} \rightarrow (\text{R}) \rightarrow \text{[BUILDING: White House]} .$$

This graph now satisfies the canonical constraints, but the relation R has not yet been defined. Background knowledge that many government officials work in the White House would lead to a conceptual graph over which a  $\lambda$ -expression for R may be defined:

$$R = (\lambda x, y) \text{ [OFFICIAL: *x]} \leftarrow (\text{AGNT}) \leftarrow \text{[WORK]} \rightarrow (\text{IN}) \rightarrow \text{[BUILDING: *y]} .$$

When this  $\lambda$ -expression is substituted for R, the following graph is obtained by  $\lambda$ -expansion:

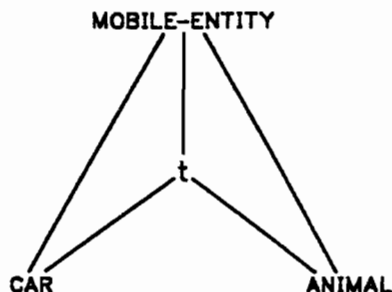


Fig. 4. New type t introduced by metaphor.

[ANNOUNCE] → (AGNT) → [OFFICIAL] ← (AGNT) ←  
 [WORK] → (IN) → [BUILDING: White House].

This graph corresponds to the phrase *An official working in the White House announced...* As this example illustrates, metonymy can be resolved by filling in unstated relationships. No new concept types are necessary.

Unlike metonymy, metaphor causes new concept types to be created. As an example, consider the metaphor *My car is thirsty*. This sentence violates the constraint that every instance of THIRSTY is an attribute of an ANIMAL:

[THIRSTY: √] ← (ATTR) ← [ANIMAL].

This metaphor cannot be interpreted by finding a missing relationship between the car and some unstated animal. Instead, the concept corresponding to the car must be linked directly to the concept [THIRSTY] by the ATTR relation, even though such a link would violate the constraint. To permit that link, Way (1991) developed a theory of *dynamic type hierarchies* that would modify the type hierarchy and change the constraints. For this example, her system would compare the types ANIMAL and CAR to find their minimal common supertype, MOBILE-ENTITY. Then a new concept type t would be introduced as a subtype of MOBILE-ENTITY and a supertype of both ANIMAL and CAR.

Figure 4 shows the place for the new type t in the hierarchy, but it does not show the differentia that distinguishes t from the supertype MOBILE-ENTITY. To define t, the system must search for properties of ANIMAL related to THIRSTY:

[THIRSTY] ← (ATTR) ← [ANIMAL] → (STAT) →  
 [REQUIRE] → (PTNT) → [LIQUID].

This graph says that thirsty is an attribute of an animal that requires liquid. The right side of the graph can serve as the body of a  $\lambda$ -expression that defines  $t$  as a mobile entity that requires liquid:

$$t = (\lambda x) [\text{MOBILE-ENTITY: } *x] \rightarrow (\text{STAT}) \rightarrow \\ [\text{REQUIRE}] \rightarrow (\text{PTNT}) \rightarrow [\text{LIQUID}] .$$

The new concept type refines the type hierarchy. Then any conceptual pattern for an ANIMAL that needs LIQUID could be generalized to  $t$ . Besides saying that a car is thirsty, one could say that it drinks, it stops for a drink, or it guzzles. After these patterns have been generalized to the type  $t$ , the standard AI techniques of inheritance would allow any MOBILE-ENTITY that requires liquid to inherit them. Trucks, planes, and ships, for example, could all be called gas guzzlers.

Different comparisons may create different supertypes for the same pair of concepts. Another metaphor that compares CAR to ANIMAL is *My car is tired*. This metaphor selects a different pattern of relations:

$$[\text{TIRE}] \leftarrow (\text{ATTR}) \leftarrow [\text{ANIMAL}] \rightarrow (\text{STAT}) \rightarrow \\ [\text{REQUIRE}] \rightarrow (\text{PTNT}) \rightarrow [\text{REST}] .$$

This graph says that tired is an attribute of an animal that requires rest. It can serve as the body of a  $\lambda$ -expression that defines  $t_2$  as a type of mobile entity that requires rest:

$$t_2 = (\lambda x) [\text{MOBILE-ENTITY: } *x] \rightarrow (\text{STAT}) \rightarrow \\ [\text{REQUIRE}] \rightarrow (\text{PTNT}) \rightarrow [\text{REST}] .$$

This new type would transfer a different family of conceptual patterns from ANIMAL to CAR: *My car is tired. I think I'll let it relax for a while. If I can afford a new one, I may even put it out to pasture.*

Dynamic type hierarchies provide a way of generating and interpreting new metaphors. They also provide insight into the origin of lexical ambiguities. When the concept type THIRSTY was generalized to a more abstract level, it became a lexical type that included its original meaning as the subtype ANIMAL-THIRSTY. The metaphorical meaning became another subtype VEHICLE-THIRSTY. If these new meanings proved to be useful, they could become word senses catalogued in dictionaries. If they did not prove to be useful, the more general sense would fall into disuse, and the concept type THIRSTY would revert to its original meaning ANIMAL-THIRSTY. The type hierarchy for SUPPORT in Figure 2 represents word senses that evolved by a similar process centuries ago. The word *support* originally meant SUPPORT-PHYS. By metaphorical extension, a new supertype was created, which is now represented by the lexical type SUPPORT. The original meaning of the word is

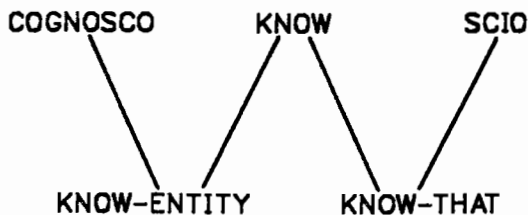


Fig. 5. Refinement of KNOW and related concept types.

only represented by the subtype SUPPORT-PHYS. The newer meanings created by metaphor are represented by other subtypes, such as SUPPORT-FIN. The type hierarchy shows the fossilized remains of long-dead metaphors. The lexical pattern in Figure 1 is a generalization of the original conceptual pattern. It is now abstract enough to include the dead metaphors as well as the etymologically first meaning.

Besides metaphor, conceptual refinement is another process that introduces polysemous patterns in the type hierarchy. Figure 5 shows an excerpt from the hierarchy with the English lexical type KNOW and the Latin lexical types COGNOSCO and SCIO. Latin uses the verb *cognosco* for knowing people and things and a separate verb *scio* for knowing that something is true. English, however, uses a single verb *know* for both senses. The deeper concept type KNOW-ENTITY is a subtype of both KNOW and COGNOSCO, and KNOW-THAT is a subtype of KNOW and SCIO. The word *know* is therefore polysemous with the two possible meanings KNOW-ENTITY and KNOW-THAT. Whereas the pattern of subtypes of SUPPORT (Figure 2) arose from old metaphors, the similar pattern of Figure 5 resulted from Latin making finer distinctions among the concept types related to KNOW.

Any concept can be refined into more specialized subtypes by making more detailed distinctions. Since different cultures may be sensitive to different features, their languages may have words that have to be translated into other languages either by rough approximations or by clumsy paraphrases. In English, for example, size is the feature that distinguishes *river* from *stream*; in French, a *fleuve* is a river that flows into the sea, and a *rivière* is either a river or a stream that flows into another river. Figure 6 shows the portion of the type hierarchy that includes the lexical types for these words and their subtypes. In translating French into English, the word *fleuve* maps into the French lexical type FLEUVE, which is a subtype of the English lexical type RIVER. There-



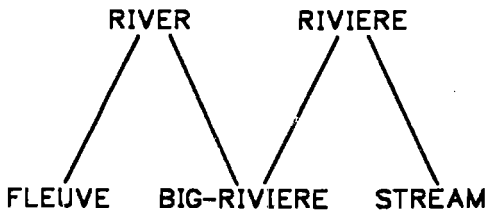


Fig. 6. Hierarchy for RIVER, STREAM, and their French equivalents.

fore, *river* is the closest one-word approximation to *fleuve*; if more detail is necessary, it could also be translated by the phrase *river that runs into the sea*. In the reverse direction, *river* maps to RIVER, which has two subtypes: one is FLEUVE, which maps to *fleuve*; and the other is BIG-RIVIERE, whose closest approximation in French is the word *rivière* or the phrase *grande rivière*. Newman (1988) used conceptual refinements like Figures 5 and 6 as a basis for developing a common lexicon for a multi-lingual machine translation system.

Even when two languages have words that are roughly equivalent in their literal meanings, they may be quite different in salience. In the type hierarchy, DOG is closer to VERTEBRATE than to ANIMAL. But since ANIMAL has a much higher salience, people are much more likely to refer to a dog as an animal than as a vertebrate. To illustrate the way salience affects word choice, Figure 7 shows part of the hierarchy that includes the English VEHICLE and the Chinese CHE. The English lexical types CAR, TAXI, BUS, TRUCK, and BICYCLE are subtypes of VEHICLE. The Chinese lexical types do not exactly match the English ones: CHE is a supertype of VEHICLE that includes TRAIN (HUOCHE), which is not usually considered a VEHICLE. QICHE includes TAXI (CHUZUQICHE) and BUS (GONGGONGQICHE) as well as CAR, which has no specific word that distinguishes it from a taxi or bus. In English, the specific words *car*, *bus*, or *taxi* are commonly used in speech, and the generic *vehicle* would normally be used only in a technical context, such as traffic laws. In Chinese, however, the word *chē* is the most common term for any kind of a vehicle. When the specific type is clear from the context, a Chinese speaker would simply say *Please call me a chē*, *I'm waiting for the 5 o'clock chē*, or *I parked my chē around the corner*. The fact that *chē* is both a stand-alone word and a component of all its subtypes enhances its salience; and the fact that *chuzuqiche* and *gonggongqiche* are four-syllable words decreases their salience. Therefore, it would sound unnatural to use the word *chuzuqiche*,