

2. Prolegomena to the Philosophy of Linguistics

Building large and comprehensive computational linguistic applications involves making many theoretical and methodological choices. These choices are made by all language processing system developers. In many cases, the developers are, unfortunately, not aware of having made them. This is because the fields of computational linguistics and natural language processing do not tend to dwell on their foundations, or on creating resources and tools that would help researchers and developers to view the space of theoretical and methodological choices available to them and to figure out the corollaries of their theoretical and methodological decisions. This chapter is a step toward generating and analyzing such choice spaces. Issues of this kind typically belong to the philosophy of science, specifically, to the philosophy of a branch of science, hence the title.

In Section 2.1, we discuss the practical need for philosophical deliberations in any credible scientific enterprise and, in particular, in our field of (computational) linguistic semantics. In Section 2.2, we discuss the reasons for pursuing theoretical work in computational linguistics. In Section 2.3, we propose (surprisingly, for the first time in the philosophy of science) definitions of what we feel are the main components of a scientific theory. In Section 2.4, we introduce a parametric space for theory building, as applied to computational linguistics. We introduce eleven basic parameters which the philosophy of science can use to reason about properties of a theory. We also discuss the relations between theories and methodologies associated with them. In Section 2.5, we extend this discussion to include practical applications of theories and their influence on relations between theories and methodologies. In Section 2.6, we illustrate the impact of choices and decisions concerning one of the 11 parameters, explicitness, for one specific theory, ontological semantics. In Section 2.7, we comment on the unusual, “post-empirical” nature of the approach to philosophy emerging from our studies and compare it to Mao’s notorious “blast furnace in every backyard” campaign.

2.1 Reasons for Philosophizing

We introduce the term “philosophy of linguistics,” similar to “philosophy of cognitive science” or “philosophy of artificial intelligence” (cf. Moody 1993: 4), to refer to the study of foundational issues of theory building in linguistics. In our view, such issues underlie and inform various important choices and decisions made in the introduction and development of certain resources (such as lexicons, ontologies and rules), of certain procedures (such as morphological, syntactic, and semantic analysis and generation), and of certain representations (such as word meaning, sentence structure, sentence meaning, etc.), as well as of the formats for all representations, rules, architectures, etc.

Less specifically to linguistics, the impetus for this work is similar to the general reasons for pursuing philosophy of science—to try to understand the assumptions, implications and other scientific, technological and societal issues and currents at work in the “object” field. The traditional philosophy of science concentrates on the “hard” sciences, mainly physics and biology. While there are contributions to the philosophy of other fields (such as the abovementioned view of cognitive science or essays on the philosophy of economics), the science of language has been largely ignored by the philosophers.⁵

Our experience of making difficult and often controversial theoretical, methodological and application-related choices in computational linguistic research made us realize how useful it would be to have a system within which to make such choices. Indeed, multiple-choice questions are really much easier than essay-like examinations! We felt the need for a basis for our decisions as well as for the alternative choices. Unfortunately, we could find no explicit choice-support framework in any of the several relevant research areas: linguistics, computational linguistics, natural language processing (NLP), artificial intelligence (AI)—of which NLP is a component— or cognitive science.

So we felt we had to venture into analyzing the theory building process in computational semantics pretty much on our own. The undertaking, we fully recognize, is risky. First, because this area is not popular. Second, because this undertaking brings us into a field, philosophy of science, of whose output we have, until now, been only consumers. Still, we see benefits to this exercise, benefits that we will try to describe below. We also hope that attempting to bring our problems to the attention of philosophers of science may underscore the need that disciplines outside the “hard sciences” have for addressing such very basic questions as a) choosing some particular theoretical constructs over others or none, b) building a hierarchy of abstract levels of representation (see also Attardo and Raskin (1991) on a solution for that in the philosophy of humor research), c) optimizing the methodology, and, most importantly, d) developing adequate justification procedures.

The research for which we first needed to answer the above questions was an application: the knowledge- and meaning-based system of machine translation called Mikrokosmos (see, for instance, Onyshkevych and Nirenburg 1994, Mahesh 1996, Viegas and Raskin 1998, Beale *et al.* 1995). The two highest-risk choices we made were the decision to go for “**deep**” **meaning analysis** (which the earlier projects in machine translation had used a great deal of hard work and ingenuity to avoid, on grounds of non-feasibility) and the decision to base our lexicon and sentence representation on a language-independent **ontology** (Mahesh 1996, Mahesh and Nirenburg 1995, Nirenburg *et al.* 1995), organized as a tangled hierarchy of frames, each of which includes and inherits a set of property slots and their fillers. Together with syntax- and non-syntax-based analysis procedures, the ontology-based lexicon (Onyshkevych and Nirenburg 1994, Viegas and Raskin 1998, Meyer *et al.* 1990) contributes to the automatic production and manipulation of text-meaning representations (TMRs—see also Carlson and Nirenburg 1990), which take the status of sentence meanings. Of course, we had to make choices of this kind in earlier implementations of ontological semantics, but it was at this time, with the development of the first large-scale application and the consequent full deployment of ontological semantics that the need became critical and practically important.

In making the choices while developing the system, we felt that we were consistently following a theory. One sign of this was that the members of the Mikrokosmos team were in agreement about

5. Of course, language itself has not been ignored by philosophy: much of the philosophy of the 20th century has been interested precisely in language. Under different historical circumstances, philosophy of language would have probably come to be known as a strain of linguistics and/or logic rather than a movement in philosophy. This would not pertain to the so-called “linguistic turn” (cf. Rorty 1967) in philosophy, a sweeping movement from the study of philosophical issues to the study of the utterances expressing these issues.

how to treat a variety of phenomena much more often than what could be expected and experienced working on other computational linguistic projects. A tempting hypothesis explaining this state of affairs was that this agreement was based on a shared implied theory. For reasons described in 4.1.3 below, we feel compelled to try and make that theory explicit.

Returning to the road map of this chapter in more specific terms, we discuss the **need** for theory in Section 2.2. Section 2.3 introduces a suggestion about what the **components** of such a theory may be like. Section 2.4 is devoted to developing a set of **parameters** for characterizing a computational linguistic theory and facilitating its comparison with other theories in the field. In Section 2.5, we give a special consideration to the important issue of the relationship between a **theory and its applications**. Section 2.6 demonstrates, on the example of ontological semantics, the impact of **choosing a certain parameter value** on the way the components of the theory are described. Section 2.7 **summarizes** our findings concerning the relationship between a science and the branch of philosophy devoted to that science.

In what follows, we will assume that the scope of analysis is computational semantics. However, we will feel free to allow ourselves to broaden this scope into theoretical linguistics when it can be done with no apparent side effects. The reason for that is that we would like to relate our work to as many other approaches as reasonable.

2.2 Reasons for Theorizing

2.2.1 Introduction: Philosophy, Science, and Engineering

The generally accepted goal of computational linguistics is the development of computational **descriptions** of natural languages, that is, of algorithms and data for processing texts in these languages. These computational descriptions are complex agglomerates, and their acquisition is an expensive and complex undertaking. The choice of a format of description as well as of the descriptive techniques is of momentous impact. The format is determined by a **theory** underlying the description. The procedures for acquiring the description constitute a **methodology**. A theory licenses a class of methodologies appropriate for description in the framework of the theory. Naturally, it is desirable to select that methodology which facilitates the most efficient production of descriptions in the theory-determined format. Figures 1-5 illustrate these definitions.

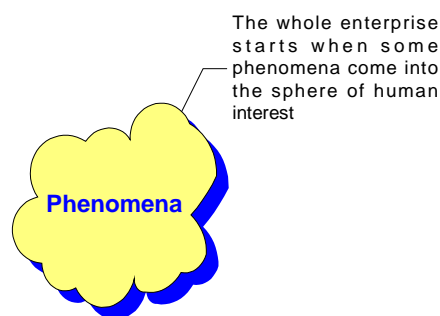


Figure 6. The most generally accepted goal of science is the description of naturally occurring phenomena.

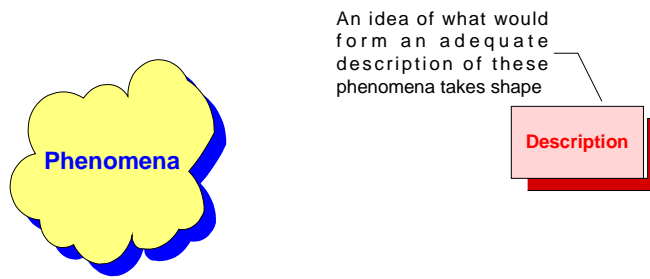


Figure 7. It is accepted that, even in the most empirical of theories, there is a step of hypothesis formation which is deductive in nature.

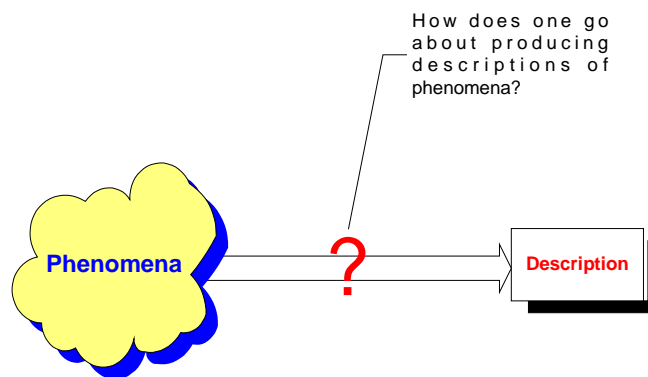


Figure 8. The need for methodology becomes clear before the need for theory.

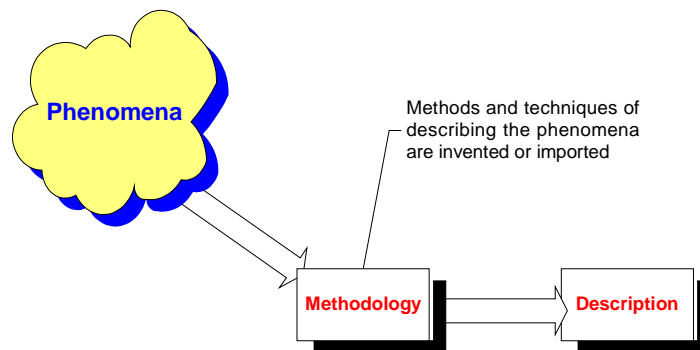


Figure 9. Methodologies can be, and often are, imported from other sciences.

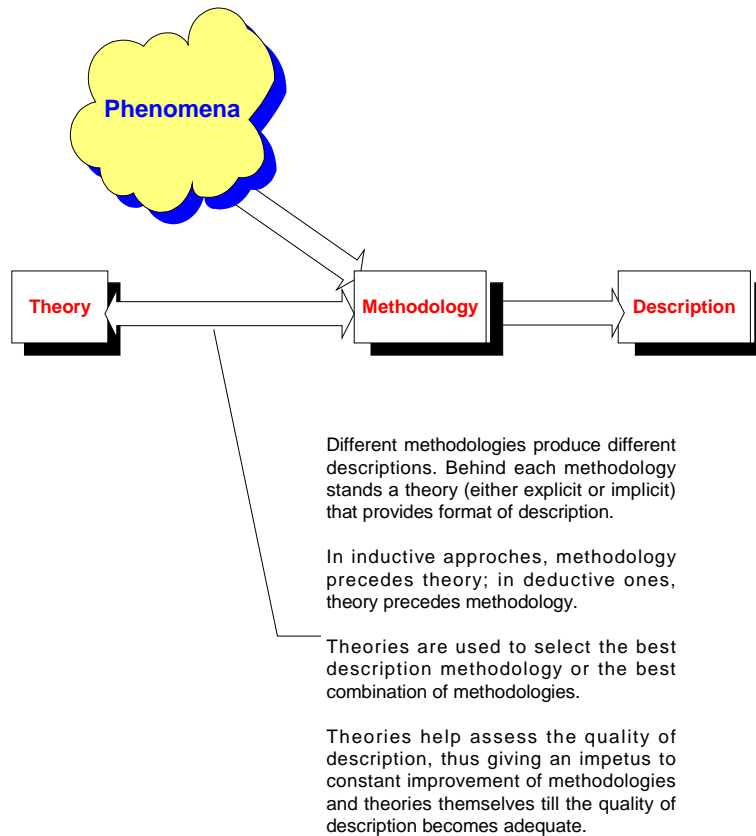


Figure 10. Relations between theories and methodologies.

The formulation of algorithms, data structures and knowledge content is a **scientific** enterprise. The optimization of acquisition methodologies and application of the complete description to a practical task is **engineering**. The formulation of the theory underlying the description must be placed in the realm of the **philosophy** of computational linguistics. The term “philosophy” is used here in the same sense as in “the philosophy of science” but differently from the sense of this term in “the philosophy of language.” Indeed, the latter has a natural phenomenon as its subject, whereas the former, similarly to computational linguistics, relates to a branch of human knowledge.

We attempt to explain and clarify these notions. In this section, we concentrate on one small but important facet of our approach, namely, the motivation for caring to develop an overt theory as the basis for computational descriptions of language. The relevant question is, *Why is it important to theorize?* This is a legitimate question, especially as a large part of the community does not seem to be overly impressed by the need to formulate the principles under which they operate.

2.2.2 Reason One: Optimization

We can put forward five (well, perhaps four and a half) reasons for overt theorizing. The first reason is that the presence of a theory, we maintain, facilitates the search for the optimum methodol-

ogy of descriptive language work, constructive engineering implementation work, formalism development, tool development, and abstract knowledge system development. If several alternative methodologies are put forward, we must have a means of preferring one over the rest. Selecting methodologies without a theoretical basis is, of course, possible and practiced, often because no reasonable theory is available. This line of action may not lead to unwelcome consequences in a particular application, as pretheoretical commonsense reasoning should not be automatically put down. Nevertheless, there is always a risk that the commonsense decisions were made based on partial evidence, which optimizes the choice of method for a partial task and says nothing about the utility of the method outside the immediate task for which it was chosen, whether within the same project or recycled in another project. Besides, the seeming absence of overt theory usually indicates the presence of an implicit, unexamined theory that guides the researcher's actions in an uncontrolled fashion, and this cannot be good.

2.2.3 Reason Two: Challenging Conventional Wisdom

The second reason pertains to what may be called the sociology of our field. (It is likely that every field follows the same pattern, but we will stick to the field we know best.) We agree with Moody (1993: 2) that “philosophy is the study of foundational issues and questions in whatever discourse (scientific, literary, religious, and so forth) they arise. The foundational issues are hard to define in a way that makes sense across all discourses.” We might add that they are equally hard to become aware of and to make explicit in any separately viewed discourse (e.g., scientific) and even within one separate discipline in that discourse (e.g., linguistics).

If a broad community of scholars shares a set of premises, and a large body of results is accumulated and accepted by this community, the need to question or even to study the original premises, statements, and rules from inside the paradigm does not appear to be pressing. This is because a scholar can continue to produce results which are significant within the paradigm as long as there remain phenomena to be described within the subject matter of that paradigm. Such a scholar, then, reasons **within** a theory but not **about** it.

There are many well-known examples of such communities (or paradigms) in 20th century linguistics: the American school of structural linguistics, roughly from Bloomfield through Harris; the school of generative grammar (which, in fact, broke into several subparadigms, such as transformational grammar (now extinct), government and binding and its descendants; lexical-functional grammar; generalized and head-driven phrase structure grammar; categorial grammar, etc.); systemic grammar, and formal semantics (see also Footnote 17 below). This “balkanization” of linguistics is taken for granted by any practitioner in the field. It is made manifest by the existence of “partisan” journals, conferences, courses and even academic programs. It is telling that in one of the preeminent academic linguistics programs in the US the sum total of one course is devoted to linguistics outside the accepted paradigm; it has been informally known for many years as “The Bad Guys.”

Of course, this state of affairs makes it more difficult for a newcomer to get a bird's eye view of our field. It also makes it more difficult for linguists to join forces with representatives of other disciplines for purposes of interdisciplinary research. Consider, as an example, how difficult it is for, say, a psychologist to form a coherent understanding of how linguistics describes meaning. The psychologist will end up with many partisan (and, often incompatible) answers to a variety of

questions, with no single paradigm proffering answers to all relevant questions.

The above state of affairs exists also in computational linguistics, though the communities here tend to be smaller, and paradigms less stable. For a computational linguist to be heard, he or she must address different paradigms, engaging in debates with people holding alternative points of view. In computational linguistics, such debates occur more or less commonly, mostly due to extra-scientific sociological reasons. The need for understanding other paradigms and the search for best arguments in a debate may lead to generalizations over the proposed alternative treatments of phenomena, to comparing the approaches and evaluating them with respect to a set of features that is acceptable to all the participants in the debate. We maintain that the grounds for such generalizations, comparison, and evaluations amount to a philosophy of the field.

Alternatively, a debate among methodological approaches to a linguistic issue can concentrate on judgments about the quality of the descriptions these approaches produce. i.e., judging a theory by the results it yields through its applications. Thus, a typical argument between two competing linguistic paradigms may focus on a complex, frequently borderline, example that one approach claims to be able to account for while claiming that the other cannot. Even in this situation, the claim rests on a notion such as descriptive adequacy (Chomsky, 1965; see also the next section), which is an important element of the underlying linguistic theory. Unfortunately, the notion of descriptive adequacy has never been clearly defined philosophically or empirically, whether as a part of the linguistic theory it must serve or as a separate theoretical entity.

2.2.4 Reason Three: Standardization and Evaluation

The third reason for theorizing is that, without a widely accepted theory, the field will resist any standardization. And standardization is essential for the evaluation of methodologies and integration of descriptions in the field. The unsurprising reason why several well-known standardization initiatives, such as the polytheoretical lexicon initiative of the mid-1980s (see, e.g., Ingria 1987), have not been as successful as one would have wanted (and as many had expected) is that standardization was attempted at the shallow level of formalism, and was not informed by similarities in theoretical statements in the various approaches being standardized.

We also believe that any evaluation of the results of application systems based on a theory should be carried out on the basis of a set of criteria external to the underlying theory. Curiously, this set of criteria in itself constitutes a theory that can be examined, questioned, and debated. Thus, the activity in the area of evaluation of machine translation and other NLP systems and resources, quite intensive in the 1990s (e.g., Ide and Veronis 1998, Arnold *et al.* 1993, King and Falkedal 1990, O'Connell *et al.* 1994), could have been viewed as an opportunity to build such a theory.

It seems that questions of theory evaluation and quality judgments about theory start to get asked only after an “initial accumulation” of data and results. A plausible picture or metaphor of the maturation of a field (Bunge 1968) is interest of its practitioners in issues of choosing high level unobservable concepts which are considered necessary for understanding explanatory mechanisms in the field. Rule-based computational linguistics has already matured to this point. Corpus-based computational linguistics may reach this point in the very near future. The explanatory mechanisms are theoretical in nature, and hence our fourth reason, or half-reason, for theorizing.

2.2.5 Reason Four: Explanation

In most cases, the subject of research is a natural phenomenon requiring an explanation. In the case of linguistics, language is that phenomenon, and it actually fronts for an even more general and mysterious phenomenon or set of phenomena referred to as the mind. A proposed theory may aspire to be explanatory but it may also choose not to do so (cf. Sections 2.4.2.2 and 2.4.3). But there can be no explanation without theory: in fact, for most users of the term theory, it is near-synonymous to explanation.

2.2.6 Reason Five: Reusability

We prefer to view concrete methodologies, tools or descriptions as instances of a class of methodologies, tools or descriptions. This will, we hope, help us to recognize a situation where an existing methodology, tool or description in some sense fits new requirements and can thus be made portable, that is, modified and used, with expectations of a considerable economy of effort. This may happen within or across applications or in the same applications for different languages.

Reusability of technology is a well-known desideratum in engineering. In our NLP work, we have made many practical decisions concerning reusability and portability of methodologies and resources. One of us strongly feels while the other suspects that the reason most of our decisions were consistent among themselves was that we have operated on the basis of a shared theory.⁶ One of the purposes for writing this book was to lay this theory out and to examine it in comparison with a number of alternatives. As Socrates said, life unexamined is not worth living.

In response to our earlier writings overtly relating descriptions and theory (see, for instance, Nirenburg *et al.* 1995, Nirenburg and Raskin 1996), some colleagues state, off the record, that they cannot afford to spend valuable resources on seeking theoretical generalizations underlying their descriptive work. This position may be justified, we believe, only when the scope of attention is firmly on a single project. Of course, one must strive for a balance between reusing methodologies and tools and developing new ones. We have discussed the issues concerning this balance in some detail in Nirenburg and Raskin (1996).

2.3 Components of a Theory

We posit that a linguistic theory (and a semantic theory or a computational linguistic theory are linguistic theories) has four components. We list them briefly with examples initially taken from one of the best-known linguistic theories, Chomsky's generative grammar. The components of the theory are: the **purview** (e.g., in generative grammar, a language *L* understood as a set of sentences and their theoretical descriptions); the **premises** (e.g., Chomsky's equating grammaticality with the intuitive ability of the native speaker to detect well-formedness of sentences in a natural language; as well as much more basic things, such as accepting the sentence as the sole unit of description or the principle that a sentence must have a unique representation for each syntactic reading it might have); the **body** (e.g., the complete list of rules in the transformational generative grammar of a language); and the **justification** statement(s) (e.g., any statements involving Chom-

6. But we have worked on this book for so long that now, on rereading this passage, we can no longer tell which of us was which.

sky's notions of descriptive and explanatory adequacy of a theory). Figures 6 and 7 illustrate these notions.

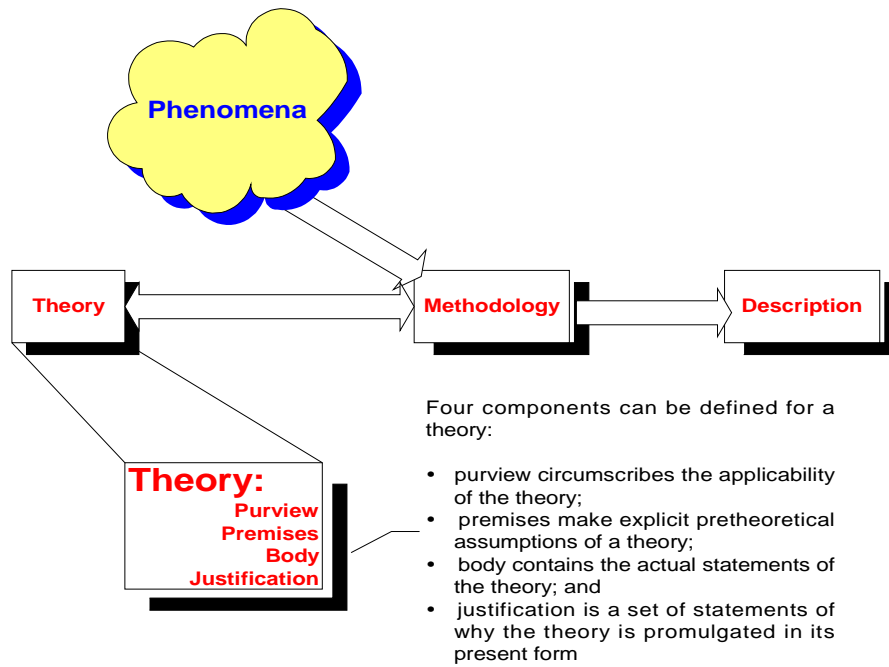


Figure 11. Components of a theory.

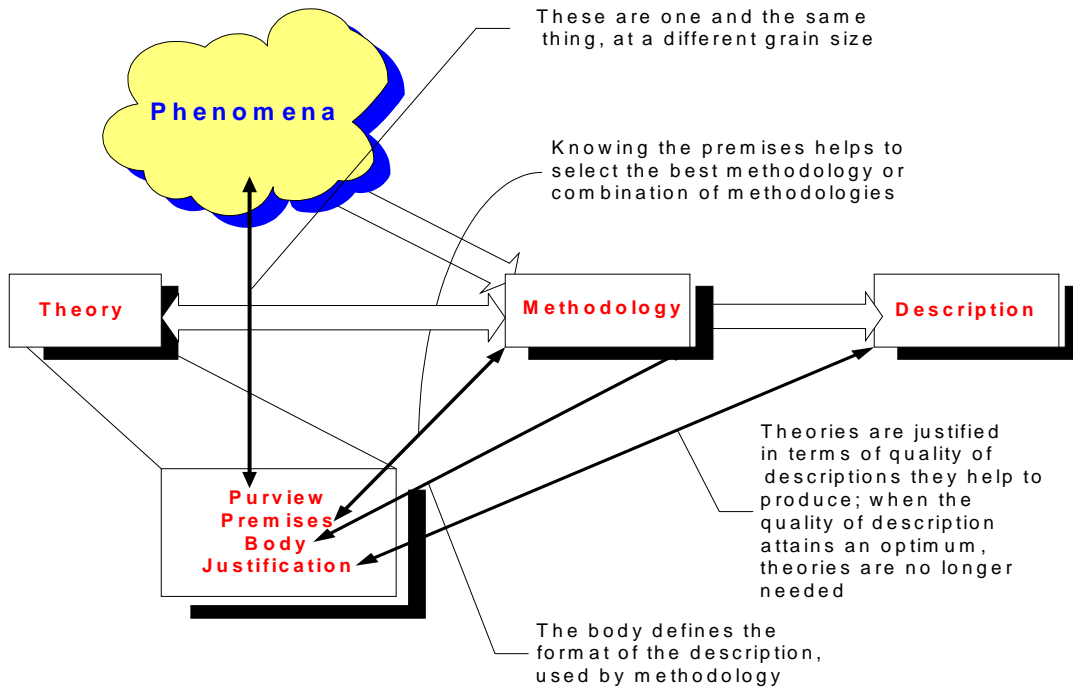


Figure 12. Relations among the components of a theory.

2.3.1 Purview

We define the **purview** (or domain) of a theory, rather a straightforward component, as the set of phenomena for which the theory is held accountable. For example, one assumes that a semantic theory will cover all meaning-related phenomena in language unless some of them are explicitly excluded. If, however, a statement billed as semantic theory is devoted only to a subset of semantics (as done with grammaticalization of meaning in, for instance, Frawley 1992—cf. Raskin 1994), without explicitly declaring this narrower purview, a misunderstanding between the reader and the author is unavoidable.

In this regard, it seems entirely plausible that some theoretical debates in our field could be rendered unnecessary if only the opponents first clarified the purviews of the theories underlying their positions. In their review of Pustejovsky (1995), Fodor and Lepore (1998) demonstrated that the purview of the theory in which they are interested intersects only marginally with the purview of Pustejovsky's generative lexicon theory. While Pustejovsky was interested in accounting for a wide range of lexical-semantic phenomena, his reviewers were content with what has become known in artificial intelligence as “upper-case semantics” (see McDermott 1978, Wilks 1999)⁷.

A recent invocation of the notion of purview can be found in Yngve (1996), where a significant amount of space is devoted to the argument that all of contemporary linguistics suffers from “domain confusion.” Yngve takes contemporary linguistics to task for not sharing the purview of the natural sciences (understood as the set of all observable physical objects) and instead operating in the “logical” domain of theoretical constructs, such as utterances, sentences, meanings, etc. rather than strings of sounds, non-linguistic behaviors accompanying those, etc.

In NLP, the purview issue takes a practical turn when a system is designed for a limited sublanguage (Raskin 1971, 1974, 1987a,b, 1990, Kittredge and Lehrberger 1982, Grishman and Kittredge 1986, Kittredge 1987). A typical problem is to determine the exact scope of the language phenomena that the description should cover. The temptation to limit the resources of the system to the sublanguage and not to account for anything outside it is strong because this approach is economical. On the other hand, such a narrow focus interferes with a possibility to port the system to an adjacent or different domain. An overt theoretical position on this issue may help to make the right choice for the situation.

2.3.2 Premises

We understand a **premise** essentially as a belief statement which is taken for granted by the theory and is not addressed in its body. A premise can be a statement like the following: Given a physical system which cannot be directly observed, if a computational system can be designed such that it produces the same outputs for the same inputs as the physical system, then the computational system must have at least some properties shared with the physical system (cf., e.g., Dennett 1979, especially, p. 60⁸). This is a formulation of the well-known “black box” model. As all premises,

7. Pustejovsky, in his response to the review (Pustejovsky 1998), commented on the differences in the premises, but those, even if valid, would be entailed by the much more essential differences in the purview.

this formulation “may be seen as concerning the most fundamental beliefs scientists as a group have regarding the nature of reality, as these beliefs are manifest in their scientific endeavors” (Dilworth 1996:1). The black box premise seems to be a version of the influential concept of supervenience in the philosophy of science, which, in its psychophysical incarnation, “is the claim that if something has a mental property at a time... then it has some physical property at that time..., such that anything with that physical property, at any time..., also has the mental property” (Zangwill 1996:68, cf. Kim 1993). In the black box premise, it is the cognitive, input-output manipulation property that is shared between the computational and the physical systems, and the goal is to determine which of the physical properties of the former, e.g., specific rules, also characterize the latter.

The notion of premise, under various names, has generated a great deal of lively debate in philosophy of science, mostly on the issues of its legitimacy and status vis-a-vis scientific theories. Dilworth (1994, 1996) refers to what we call premises as presuppositions or principles and states that “they cannot have been arrived at through the pursuit of science, but must be, in a definite sense, pre-scientific, or metascientific” (1996:2). Moody (1993: 2-3) refers to what we call premises as foundational issues, which he defines as presuppositions of scientific statements, such as “there are sets” in mathematics.⁹ A crucial point here is that this latter fact should not preclude careful examination of a theory’s premises. Many philosophers, however, adhere to the belief (which we consider a *non sequitur*) that the premises of a theory cannot be rigorously examined specifically because of their metascientific nature (cf. Davidson 1972; for Yngve, no subject-related premises are acceptable in the pursuit of any science (1996: 22)).

Premises seem to play the same role as axioms in algebraically defined theories, except that the latter are explicitly stated and, thus, included in the body of the theory. An axiom is the starting point in truth-preserving derivation of propositions in the body of a theory. A premise participates in such a derivation implicitly. This is why it is rather difficult to explicate the premises of a theory and why theorists often find this task onerous.

Whether they are explicated or implicit, premises play a very important role in scientific work. Just as specifying the purview of a theory establishes the boundaries of the phenomena to be accounted for by that theory, so the premises of a theory determine what questions it should address and what statements would qualify as satisfactory answers to these questions. In this sense, premises can be said to define “the rules of the scientific game.” One such important rule is

8. In the philosophy of science, especially of AI, Marr’s (1982) approach to computational modeling of vision seriously influenced the thinking and writing on computationalism in general, i.e., stages, validity, and goals of top-down computer models for mental processes and activity, mostly on the strong-AI premise (cf. Sections 2.4.2.2 and 2.4.3 below). For various foundational issues with the approach, see Kitcher (1988), Dennett (1991), Horgan and Tienson (1994: 305-307), Hardcastle (1995), Gilman (1996).

9. “It would be fair to say,” Moody explains (1993: 2), “that the foundations of a discipline are the concepts and principles most taken for granted in it. In mathematics, for example, a foundational proposition would be, “There are sets.” The philosopher of mathematics might or might not want to deny this proposition, but would certainly want to ask in a rigorous way what it means.” Setless mathematics is definitely a possibility, even if it has not and will not be explored, but, as we will see in Section 2.6.2 below, the computational linguistic premises we have to deal with sound less trivial and impose tough and real choices for the researcher.

defining what is meant by completion, closure, or success of a theory. Thus, a scientific theory based on the black box premise is complete, closed, and successful if it succeeds in proving that at least some properties of the computational system are shared by the physical system. In the absence of such a premise, the computational modeling of physical systems would not have a theoretical standing.

The need for overt specification of premises became clear to us when we tried to understand what was funny in the passage from Jaroslav Hašek's (1974: 31) "The Good Soldier Švejk," where a mental asylum patient is reported to believe that "inside the globe there was another globe much bigger than the outer one." We laughed at the absurdity of the notion because of the unsaid premise that no object can contain a larger one. Trying to prove the falsity of the original statement, we reduced the problem to the linear case of comparing two radii. At this point, we realized that no proof was available for the following statement: "when superimposed, a longer line fully incorporates a shorter line." This statement seems to be taken for granted in elementary geometry. This is an example of a premise. It is possible that any premise could in principle be formalized as an axiom. An axiom is a premise made explicit. And we believe that any theory would profit from making all of its premises into axioms.

2.3.3 Body

The **body** of a theory is a set of its statements, variously referred to as laws, propositions, regularities, theorems or rules. When still unimplemented, the body of a theory amounts to a statement about the **format** of the descriptions that are obtainable using this theory. When a theory is implemented, its body is augmented by **descriptions**. In fact, if one assumes the possibility of attaining a closure on the set of descriptions licensed by the theory, at the moment when this closure occurs, the theory loses the need for the format: the body of the theory will simply contain the set of all the descriptions. This, of course, is a strong idealization for a semantic theory, as the set of all utterances that may require meaning representation (that is, description) is, like Denny's restaurants, always open.

An interesting relation exists between the premises and the body of the theory, that between the ideal and the actual. We agree with Dilworth (1996: 4) that premises (which he terms 'principles') "constitute the core rather than the basis of science in that they are not general self-evident truths from which particular empirical truths can be formally deduced, but are rather ideal conceptions of reality which guide scientists' investigations of actual reality. From this perspective, what makes a particular activity scientific, is not that the reality it uncovers meets the ideal, but that its deviation from the ideal is always something to be accounted for." While the premises of a theory induce an ideal realization in the body, the actual set of propositions contained in the latter, at any given time of practical research, accounts only for a part of the ideal reality and in less depth than the ideal realization expects. Besides being generally true, it has the practical consequence of clarifying the relation between the ideal methodology required by the ideal body and the practical methodologies developed and applied both in theoretical and especially in applicational work (see Section 2.5 below). We will also see there that, in the process of description, not only our methodologies but also the body of the theory will undergo a change as we improve our guesses concerning the elements of the ideal body.

2.3.4 Justification

The concern for **justification** of theories is relatively recent in the scientific enterprise. For centuries, it was accepted without much second thought that direct observation and experiments provided verification or disproof of scientific hypotheses. It was the logical positivists (see, for instance, Carnap 1936-1937, 1939, 1950, Tarski 1941, 1956, Reichenbach 1938, Popper 1959, 1972, Hempel 1965, 1966; see also Braithwaite 1955, Achinstein 1968), whose program centrally included the impetus to separate science from non-science, that carefully defined the notion of justification and assigned it a pivotal place in the philosophy of science. In fact, it would be fair to say that the philosophy of science as a field emerged owing to that notion.

Justification is the component of a theory which deals with considerations about the quality of descriptions and about the choices a theory makes in its premises, purview and body. “How is the reliability of our knowledge established?” This is a standard question in contemporary scientific practice. “All theories of justification which pose this question, must divide the body of knowledge into at least two parts: there is the part that requires justification [that is, our premises, purview and body] and there is the part that provides it [that is, our justification]” (Blachowicz 1997:447-448).

The process of justifying theories intrinsically involves a connection between the realm of theoretical statements and the directly observable realm of concrete experiences. The critical and highly divisive issue concerning justification is the status of unobservable constructs and, hence, of disciplines outside the traditional natural sciences, that is, fields such as sociology, psychology, economics, or linguistics, where, even in the best case, empirical observation is available only partially and is not always accessible. Besides, serious doubts have gradually emerged about the status of direct observation and its translatability into theoretical statements. An influential opinion from computer science has advocated a broader view of experiment and observation: “Computer science is an empirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method. Nonetheless, they are experiments” (Newell and Simon 1976: 35-36).

First, Popper’s work (1959, 1972) firmly established the negative rather than positive role of empirical observation. The novelty of the idea that direct observation can only falsify (refute) a hypothesis (a theory) and never verify (confirm) it had a lasting shocking effect on the scientific community. The matter got more complicated when Quine (1960, 1970) expressed serious doubts concerning the feasibility of the related problem of induction, understood as the ability of the observer to translate direct experience into a set of statements (logical propositions for the positivists) that constitute scientific theories. Once the logical status of observation was withdrawn, it has lost its attraction to many philosophers. According to Haack (1997:8), “[a] person’s experiences can stand in causal relation to his belief-states but not in logical relation to the content of what he believes. Popper, Davidson, Rorty *et al.* conclude that experience is irrelevant to justification”—see, for instance, Popper (1959, 1972), Davidson (1983, 1984, 1987), Rorty (1979, 1991)¹⁰; cf. Haack (1993). In other words, direct experience may confirm or disconfirm a person’s belief but does nothing to the set of logical propositions describing his belief system. Moreover, the modern approach to justification “rejects the idea of a pretheoretical observation vocabulary: rather, it is our scientific theories themselves that tell us, in vocabulary which is inev-

itably theory-laden, what parts of the world we can observe” (Leeds 1994:187).

What is, then, the status of theoretical constructs and statements which are unobservable in principle? Orthodox empiricism continues to deny any truth to any such statements (cf. Yngve 1996). Constructive empiricism (e.g., van Fraassen 1980, 1989) extends a modicum of recognition to the unobservable, maintaining that “we should believe what our best theories tell us about observables—that is, about the observable properties of observable objects—by contrast, we should merely accept what our theories tell us about the in principle unobservable, where accepting a theory amounts to something less than believing it” (Leeds 1994:187). Realists make one step further: “The hallmark of realism (at least as Dummett understands it) is the idea that there may be truths that are even in principle beyond the reach of all our methods of verification” (Bar-On 1996:142; cf. Dummett 1976 and especially 1991).

A moderate hybrid view that has been recently gaining ground combines foundationalism, a mild version of empiricism, with coherentism, a view that places the whole burden of justification on the mutual coherence of the logical propositions constituting a theory—see Haack (1993, 1997; cf. Bonjour 1997). According to this view, a scientific theory consists of two kinds of propositions: those that can be verified empirically and those which are unverifiable in principle. The former are justified in the empiricist way. The latter, on the ground of their internal coherence as well as their coherence with the empirically justified propositions.¹¹ And all the propositions, no matter how they are justified, enjoy equal status with regard to the truth they express.

The dethroning of empirical observation as the privileged method of justification reaches its apotheosis in the view that the depth and maturation of every science requires the introduction and proliferation of an increasingly elaborate system of unobservable theoretical concepts. This view has even been applied to physics, a traditional playground of the philosophy of science. The extent to which a science is sophisticated and useful is measured in its success to move “from data packages to phenomenological hypotheses, to mechanism hypotheses,” (Bunge 1968:126-127) where the last depend entirely on a complex hierarchy of untestable theoretical concepts.

How is justification handled in current efforts in linguistics and natural language processing? There is a single standard justification tool, and it is Popperian in that it is used to falsify theories. Not only is this tool used as negative justification; surprisingly, it also serves as a *de facto* impetus to improve theories. We will explain what we mean here on an example. In generative grammar, a proposed grammar for a natural language is called descriptively adequate if every string it generates is judged by the native speaker as well-formed, and if it generates all such strings and nothing but such strings. A grammar is a set of formal rules. Standard practice to make a contribution in the field is to find evidence (usually, a single counterexample) in language that the application of a rule in a grammar may lead to an ill-formed string. This finding is understood as refutation of the grammar which contains such a rule. The next step, then, is to propose a modification or substitution of the offending rule with another rule or rules which avoids this pitfall. An extension of this principle beyond purely grammatical well-formedness as a basis of justification to include

10. This group of “anti-experientialists” is not homogeneous: Rorty stands out as accused of an attempt “to discredit, or to replace, the whole analytical enterprise” (Strawson 1993), something neither Popper nor Davidson are usually charged with.

presupposition, coherency, and context was tentatively suggested in Raskin (1977b—see also 1979, 1985, and Section 6.1).

The Popperian justification tool for linguistics leaves much to be desired. First of all, it is best suited to address a single rule (phenomenon) in isolation. This makes the application of the tool a lengthy and impractical procedure for the justification of an entire grammar. Secondly, according to at least one view in the philosophy of language, “justification is gradational” (Haack 1997:7) and must, therefore, allow for quality judgments, that is, for deeming a theory better than another rather than accepting or rejecting an individual theory. On this view, the above tool is not a legitimate justification tool.

Little has been written on the issue of justification in linguistics. It is not surprising, therefore, that in the absence of a realistic set of justification criteria, the esthetic criteria of simplicity, elegance and parsimony of description were imported into linguistics by Chomsky, apparently, from a mixture of logic and the philosophy of science, (1957: 53-56; 1965: 37-40) and thereafter widely used to support judgments about grammars. Moreover, once these notions were established in linguistics, they received a narrow interpretation: simplicity is usually measured by the number of rules in a grammar (the fewer rules, the simpler the grammar), while brevity of a rule has been interpreted as the measure of the grammar’s elegance (the shorter the rule, the more elegant the theory).¹² Parsimony has been interpreted as resistance to introducing new categories into a grammar.¹³

2.4 Parameters of Linguistic Semantic Theories

In this section, we attempt to sketch the conceptual space within which all linguistic semantic theories can be positioned. This space is composed of diverse parameters. Each theory can be charac-

11. Chomsky assumes a similar position as early as *Syntactic Structures* (1957: 49) without any indication that this is a very controversial issue: “A grammar of the language L is essentially a theory of L,” he writes. “Any scientific theory is based on a finite number of observations, and it seeks to relate the observed phenomena and to predict new phenomena by constructing general laws in terms of hypothetical constructs such as (in physics, for example) “mass” and “electron.” Similarly, a grammar of English is based on a finite corpus of utterances (observations), and it will contain certain grammatical rules (laws) stated in terms of particular phonemes, phrases, etc., of English (hypothetical constructs). These rules express structural relations among the sentences of the corpus and the indefinite number of sentences generated by the grammar beyond the corpus (predictions). Our problem is to develop and clarify the criteria for selecting the correct grammar for each language, that is, the correct theory of this language.” But Chomsky took his position much further by letting the theory itself decide certain matters of reality: “Notice that in order to set the aims of grammar significantly it is sufficient to assume a partial knowledge of sentences and non-sentences. That is, we may assume for this discussion that certain sequences of phonemes are definitely sentences, and that certain other sequences are definitely non-sentences. In intermediate cases we shall be prepared to let the grammar itself decide, when the grammar is set up in the simplest way so that it includes the clear sentences and excludes the clear non-sentences” (*op. cit.*: 13-14). If this sounds too radical—letting a theory decide if a string is a sentence or not, which is a matter of empirical fact for the speaker—he defers to an authoritative source: “To use Quine’s formulation, a linguistic theory will give a general explanation for what ‘could’ be in language on the basis of ‘what is plus simplicity of the laws whereby we describe and extrapolate what is’ (... Quine [1953:] 54)” (*op. cit.*: 14fn.). Chomsky’s position on justification has never changed, largely because the issue itself was essentially abandoned by him after 1965.

terized by a particular set of parameter values. This provides each theory with a perspective on a number of choices made in it. This exercise is helpful because, in building theories (as, we should add, in everything else, too), people tend to make many choices unconsciously and to be often unaware of their existence. Awareness of one's options is a good start toward creating better theories. Creating the choice space for theories in natural sciences is a job for the philosophy of science. For linguistic theories, it is, therefore, the responsibility for the philosophy of linguistics.

We will list a number of dimensions, parameters and values for characterizing and comparing semantic theories. The parameters can be grouped somewhat loosely along a variety of dimensions, namely, "related to theory itself," "related to methodology induced by the theory," "related to status as model of human behavior" (e.g., mind, language behavior, etc.), "related to internal organization" (e.g., microtheories).

2.4.1 Parameters Related to Theory Proper

In this section, we focus on the theoretical parameters of adequacy, effectiveness, explicitness, formality, and ambiguity.

2.4.1.1 Adequacy

A theory is adequate if it provides an accurate account of all the phenomena in its purview. Adequacy can be informally gauged through introspection, by thinking, to take an example of linguistics, of additional language phenomena that a particular theory should cover. It can be established rule by rule using the standard linguistic justification tool discussed in Section 3.4 above. There is an additional mechanical test for adequacy in computational linguistics, namely determining whether a description helps to solve a particular problem, such as figuring out syntactic dependencies inside a noun phrase. As demonstrated by Popper's (1959, 1972; see also Section 2.3.4 above), it is much easier to demonstrate that a theory is inadequate than to establish its adequacy.

Our definition of adequacy refers to an ideal case. As mentioned in Section 2.3.3 above, linguistic theories are never quite complete (as a simple example, consider that no dictionary of any lan-

-
12. This is, of course, a rather cavalier use of the very complex and controversial category. Chomsky does not refer to a vast field of study with regard to the category—see, for instance, Popper (1959—cf. Simon 1968: 453), Good (1969) (recanted in Good 1983), Rosenkrantz (1976), and especially Sober (1975). According to Richmond (1996), simplicity was attempted to be explained by these authors in terms of the equally complex categories of familiarity and falsifiability, content, likelihood, and relative informativeness.
 13. Especially in applications but to some degree also in theoretical work, one must be careful not to yield to the temptation to "show one's ingenuity" by introducing properties, categories and relations that might be descriptively adequate but that are not necessary for description. Such a proliferation of terms is generally typical of structuralist work, from which the concern of both the generative and computational approaches about the use of each category in a rule or an algorithm is typically missing—for a contemporary structuralist example (see Mel'čuk 1997, 1998). An example of the use of parsimony in transformational grammar (most influentially, perhaps, Postal 1971) has been the long practice of never introducing a new transformational rule for describing a phenomenon unless that rule could be independently motivated by its applicability to a different class of phenomena. In our own work, we discovered that the distinction between the attributive and predicative use of adjectives, widely considered essential, has no bearing on a theory of adjectival meaning and should not therefore be included in it (for details, see Raskin and Nirenburg 1995, 1998)—for lack of any independent motivation.

guage can guarantee that it includes every word in the language). In practice, the parameter of adequacy is applied to theories which have accounted correctly for the phenomena they have covered, i.e., for their purviews. One can only hope that the theory will remain adequate as it extends its purview to include new phenomena.

2.4.1.2 Effectiveness

We will call a theory *effective* if we can show that there exists, in principle, a methodology for its implementation. We will call a theory *constructive* if a methodology can be proposed that would lead to an implementation of a theory in finite time.¹⁴ Let us first illustrate the above distinction as it pertains not to a linguistic theory but to a readily formalizable theory describing the game of chess. We will discuss this problem in our own terms and not those familiar from game theory. For our purposes here, we will use the game of chess as an objective phenomenon for which theories can be propounded.¹⁵ Theories that can be proposed for chess include the following three competing ones: “White always wins,” “Black always wins,” or “Neither White or Black necessarily wins.” An early theorem in game theory proves that the first of these theories is, in fact, the correct one, namely that there is a winning strategy for White. This means several important things: first, that it is possible, in principle, to construct an algorithm for determining which move White must make at every step of the game; second, because the number of possible board positions is finite (though very large), this algorithm is finite, that is, it will halt (there is a rule in chess that says that if a position repeats three times, the game is a draw); third, this algorithm has never been fully developed. Mathematical logic has developed terminology to describe situations of this kind: the first fact above makes the theory that White always wins *decidable* (alternatively, one can say that the problem of chess is *solvable* for White). The third fact says that this theory has not been proven *computable*.

The following is a formal definition of decidability of a theory or of the solvability of a problem in it: “The study of *decidability* involves trying to establish, for a given mathematical *theory T*, or a given problem *P*, the existence of a decision algorithm AL which will accomplish the following task. Given a sentence *A* expressed in the language of *T*, the algorithm AL will determine whether *A* is true in *T*, i.e., whether $A \in T$. In the case of a problem *P*, given an instance *I* of the problem *P*, the algorithm AL will produce the correct answer for this instance. Depending on the problem *P*, the answer may be “yes” or “no”, an integer, etc. If such an algorithm does exist, then we shall variously say that the *decision problem* of *T* or *P* is *solvable*, or that the theory *T* is *decidable*, or simply that the problem *P* is solvable. Of AL we shall say that it is a decision procedure for *T* or *P*” (Rabin 1977: 596; see also Uspenskiy 1960 on the related concepts of decidable and solvable sets).

Establishing the existence of the algorithm and actually computing it are, however, different matters: the mere existence of the algorithm makes the theory *decidable*; the actual demonstration of

14. The notion of finiteness brings up the dimension of theory implementation within given resources. We will discuss this issue in detail in Section 2.5 below.

15. Searle (1969) denies chess the status of objective reality because, unlike natural phenomena whose laws may be discovered, chess is a constructed phenomenon “constituted” by its rules. For our purposes, this distinction is immaterial; see a more detailed discussion of the relation of linguistic theories to reality in Nirenburg and Raskin (1996).

the algorithm makes it *computable*. There is also the matter of *practical*, as opposed to *theoretical*, decidability (and computability): “[w]ork of Fischer, Meyer, Rabin, and others has... shown that many theories, even though decidable, are from the practical point of view undecidable because any decision algorithm would require a practically impossible number of computation steps” (Rabin 1977: 599¹⁶). The above pertains to the second fact about the theory that White always wins: in some cases, the decision algorithm for a theory is infinite, that is, it does not halt; this is not the case of the chess theory in question; however, this may not make this theory practically decidable—or machine tractable—because the complexity requirements of the decision algorithm may exceed any available computational resources.¹⁷

The logical notions of decidability and computability work well for the example of chess but are not applicable, as defined, to linguistic theory, because language is not, strictly speaking, a mathematical system. It is precisely because of the fact that linguistic theories cannot be completely and neatly formalized that we first introduced the concepts of effectiveness and constructiveness to avoid using the more narrowly defined parallel pair of terms ‘decidability’ and ‘computability,’ respectively, outside of their intended mathematical purview. Many of the procedures used in developing linguistic theories are, therefore, difficult to automate fully. For example, in ontological semantics, description (namely, the acquisition of static and dynamic knowledge sources) is semi-automatic in a well-defined and constraining sense of using human intuition (Mahesh 1996, Viegas and Raskin 1998—see also Section 2.5 below). The human acquirers are assisted in their work by specially designed training materials. These materials contain guidance of at least two kinds, how to use the tools and how to make decisions. Statements about the latter provide a very good example of the part of the theory which is not formal.

Contemporary linguistic theories aspire to exclude any recourse to human participation except as a source of typically uncollected judgments about the grammaticality of sentences. But there is a steep price to pay for this aspiration to full formality, and this statement seems to hold for computational linguistics as well. It is fair to say that, to-date, fully formalizable theories have uniformly been of limited purview. Formal semantics is a good example: in it, anything that is not formalizable is, methodologically appropriately, defined out of the purview of the theory (Heim and Kratzer 1998 is a recent example, but see also Frawley 1992; cf. Raskin 1994): for example, the study of quantification, which lends itself to formalization, has been a central topic in formal semantics, while word sense definition that resists strict formalization is delegated to a sister discipline, lexical semantics. The proponents of full formalization inside lexical semantics continue with the purview-constraining practices in order to remain fully formal. In contrast, still other linguistic theories, ontological semantics among them, have premises that posit the priority of phenomenon coverage over formalization in cases of conflict; in other words, such theories decline to limit their purview to fit a preferred method.

These not entirely formal¹⁸ theories would benefit the most from a study of the practical consequences of their being constructive; effective but not constructive, or neither effective nor con-

16. For early seminal work on (un)decidability, see Tarski (1953). For further discussion, see Ackermann (1968) and Gill (1990). On computability, see a good recent summary in Ershov (1996). On constructibility, very pertinently to our effectiveness, see Mostowski (1969) and Devlin (1984). On decidability in natural language, cf. Wilks (1971).

structive.

Ontological semantics can be presented as a theory producing descriptions of the form $M_S = TMR_S$, i.e., the meaning M of a sentence S in a natural language L , e.g., English, is represented by a particular formal text-meaning representation (TMR) expression. In each implementation of ontological semantics, there is an algorithm, the analyzer, for determining the truth of each expression in the above format: it does that by generating, for each sentence, its unique TMR. This establishes the constructiveness of the theory (as well as its effectiveness) *post hoc*, as it were. We will discuss what, if anything, to do with a theory which is known not to be constructive in 2.4.2.1 below.

2.4.1.3 Explicitness

Theories overtly committed to accounting in full for all of their components are explicit theories. In Section 2.6 below, we illustrate this parameter on the specific example of ontological semantics. Explicitness has its limits. In a manner akin to justification, discussed above, and all other components and parameters of theories, explicitness is “gradational.” Somewhat similarly to the situation with adequacy, explicitness is an ideal notion. A theory which strives to explicate all of its premises, for instance, can never guarantee that it has discovered all of them—but we believe that, both theoretically and practically, one must keep trying to achieve just that; this is, basically, what this chapter is about.

-
17. Will, for instance, a syntactic analyzer based on Chomsky’s transformational grammar (TG) be tractable? For the sake of simplicity, let us assume its output for each sentence to be a representation, with each constituent phrase bracketed and labeled by the rules that generated the phrase. The input to the analyzer will be simply a string of words, and the analyzer will have to insert the parentheses and labels. Is it computable? It is, but only for grammatical sentences. If the string is ungrammatical, the algorithm will never find a sequence of rules, no matter how long, that will generate the string and will continue to attempt the derivation indefinitely. The uncomputability of the system, if not supplied with an external halting condition, is the killing argument against the TG formalism as the basis for computational syntactic analysis (parsing). Apparently, no such halting condition could be formulated, so a high-powered effort to develop such an analyzer for TG failed (see about the MITRE project in Zwicky *et al.* 1965 and Friedman 1971), as did, in fact, a similar effort with regard to Montague grammars (Hobbs and Rosen-schein 1977, Friedman *et al.* 1978a,b, Hirst 1983, 1987; cf. Raskin 1990: 117). In fact, a kind of natural selection occurred early on, when NLP systems started selecting simpler and possibly less adequate grammatical models as their syntactic bases (see, for instance, Winograd 1971, which deliberately uses a simplified version of systemic grammar—see Berry 1975, 1977, Halliday 1983; cf. Halliday 1985—rather than any version of transformational grammar), and later, several more tractable and NLP-friendly approaches, such as head phrase structure grammar (Pollard 1984, Pollard and Sag 1994), tree-ad-joining grammars (Joshi *et al.* 1975, Joshi 1985, Weir *et al.* 1986), or unification grammars (Kay 1985, Shieber 1986) were developed. NLP-friendliness does not mean just an aspect of formality—it has also to do with literal friendliness: Chomsky’s open hostility to computation in linguistics as manifested most publicly in “The Great Debate,” aka “the Sloan money battle,” mostly by proxy, between Chomsky and Roger Schank (Dresher and Hornstein 1976, 1977a,b, Schank and Wilensky 1977, Winograd 1977; for a personal memoir, see Lehnert 1994: 148ff; for a related discussion, see Nirenburg 1986), has contributed greatly to the practical exclusion of Chomsky’s grammars, from standard theory (Chomsky 1965) to extended standard theory (Chomsky 1971) to traces (Chomsky 1973) to government and binding and principles and parameters (Chomsky 1981) and, most recently, to the minimalist position (Chomsky 1995).

2.4.1.4 Formality and Formalism

A theory can be formal in two senses. Formality may mean completeness, non-contradictoriness and logically correct argumentation. It may also refer to the use of a mathematical formalism. The two senses are independent: thus, a theory may be formal in both senses, in either sense, or in neither.

Formality in the second sense usually means a direct application of a version of mathematical logic, with its axiomatic definitions, theorems, in short, all its well-established formal derivation machinery, to a particular set of phenomena.¹⁹ The formalism helps establish consistency of the set of statements about the phenomena. It also establishes relations of equivalence, similarity, proximity, etc., among terms or combinations of terms and through this, among the phenomena from the purview of the theory that the logic formalizes. This may result in the imposition of distinctions and relations on the phenomena which are not intuitively clear or meaningful. Wilks (1982: 495) has correctly characterized the attempts to supply semantics for the formalism in order to apply it to NLP, as an “appeal to external authority: the Tarskian semantics of denotations and truth conditions for some suitably augmented version of the predicate calculus (Hayes, 1974; McDermott, 1978).”

A danger of strict adherence to formality in the sense of formalism is the natural desire to remove from theoretical considerations phenomena which do not lend themselves to formalization using the formal language of description. This, in turn, leads to modifications in the purview of a theory and can be considered a natural operation. Indeed, modern science is usually traced back to Galileo and Newton, who made a departure from the then prevalent philosophical canon in that they restricted the purview of their theories to, very roughly, laws of motion of physical bodies for the former and physical forces for the latter. By doing so, they were able to make what we now accept as scientific statements about their purviews. The crucial issue is the ultimate utility of their theories, even if their purviews were narrower than those of other scholarly endeavors.

18. The issue of computation on the basis of a theory which is not completely formal is very complex. The content of Nirenburg and Raskin (1996), Mahesh (1996), and Viegas and Raskin (1998) can be considered as a case study and illustration of this issue.

19. Quine (1994: 144) puts it very simply and categorically: “On the philosophical side, the regimentation embodied in predicate logic has also brought illumination quite apart from the technology of deduction. It imposes a new and simple syntax on our whole language, insofar as our logic is to apply. Stripped down to the austere economy that I first described for predicate logic, our simple new syntax is as follows. The parts of speech are: (1) the truth-functional connective, (2) the universal quantifier, (3) variables, and (4) atomic predicates of one and more places. The syntactic constructions are: (1) application of a predicate to the appropriate number of variables to form a sentence; (2) prefixure of a quantifier, with its variable, to a sentence; and (3) joining sentences by the truth-functional connective and the adjusting parentheses. I hesitate to claim that this syntax, so trim and clear, can accommodate in translation all cognitive discourse. I can say, however, that no theory is fully clear to me unless I can see how this syntax would accommodate it. In particular, all of pure classical mathematics can be thus accommodated. This is putting it mildly. The work of Whitehead and Russell and their predecessors and successors shows that the described syntax together with a single two-place predicate by way of extra-logical vocabulary, namely the ‘e’ of class membership, suffices in principle for it all. Even ‘=’ is not needed; it can be paraphrased in terms of ‘e’.”

Turning back to our own case, we must consider the trade-off in computational linguistics between limiting the purview of a theory and keeping it potentially useful. Our own attempts to alleviate this tension have found their expression in the concept of microtheories (see Chapter 1, also see, for instance, Raskin and Nirenburg 1995), though that, in turn, leads to the still open issue of how, if this is at all possible, to make these microtheories coexist without contradictions.

In an alternative approach, formalism is not the impetus for description but rather plays a supporting role in recording meaningful statements about the phenomena in the purview of the theory. In other words, in this approach, content is primary and formalism, secondary. Ontological semantics has been developed on this principle. There is room for formalism in it: TMRs are completely formal, because they are defined syntactically using an explicit grammar, represented in Backus-Naur form (BNF) and semantically by reference to a constructed ontology. The TMR formalism has been determined by the content of the material that must be described and by the goals of the implementations of ontological semantics. In an important sense, the difference between the two approaches is similar to that between imposing the formalism of mathematical logic on natural language and the short-lived attempts to discover “natural logic” the “inherent” logic underlying natural language (McCawley 1972, Lakoff 1972). While natural logic was never developed or applied, it provided an important impetus for our own work by helping us to understand that formality is independent of a formalism. In this light, we see the structures of ontological semantics as expressing what the natural logic movement could and should have contributed.

Going back to that first sense of formality, we effectively declare it a necessary condition for a theory and do not consider here any theories that do not aspire to be formal in that sense. In practice, this kind of formality means, among other things, that all terms have a single meaning throughout the theory, that there can be no disagreement among the various users about the meaning of a term or a statement, that each phenomenon in the purview is characterized by a term or a statement, and that every inference from a statement conforms to one of the rules (e.g., *modus ponens*) from a well-defined set.

We believe that the best result with regard to formality is achieved by some combination of formalism importation and formalism development. For instance, an imported formalism can be extended and modified to better fit the material. It might be said that this is how a variety of specialized logics (erotetic logic, modal logic, deontic logic, multivalued logic, fuzzy logic, etc.) have come into being. Each of these extended the purview of logic from indicative declarative utterances to questions, modalities, expressions of necessity, etc.

The idea of importing a powerful tool, such as a logic, has always been very tempting. However, logical semantics was faulted by Bar Hillel, himself a prominent logician and philosopher, for its primary focus on describing artificial languages. Bar Hillel believed that treatment of meaning can only be based on a system of logic: first, because, for him, only hypotheses formulated as logical theories had any scientific status and, second, because he believed that inference rules necessary, for instance, for machine translation, could only be based on logic. At the same time, he considered such logical systems *unattainable* because, in his opinion, they could not work directly on natural language, using instead one of a number of artificial logical notations. “...The evaluation of arguments presented in a natural language should have been one of the major worries... of logic since its beginnings. However,... the actual development of formal logic took a dif-

ferent course. It seems that... the almost general attitude of all formal logicians was to regard such an evaluation process as a two-stage affair. In the first stage, the original language formulation had to be rephrased, without loss, in a normalized idiom, while in the second stage, these normalized formulations would be put through the grindstone of the formal logic evaluator.... Without substantial progress in the first stage even the incredible progress made by mathematical logic in our time will not help us much in solving our total problem” (Bar Hillel 1970: 202-203).

2.4.1.5 Ambiguity

This parameter deals with the following issue: Does the theory license equivalent (synonymous, periphrastic) descriptions of the same objects? On the one hand, it is simpler and therefore more elegant to allow a single description for each phenomenon in the purview, in which case the issue of alternative descriptions and their comparison simply does not arise. However enticing this policy might be, it is difficult to enforce in practice. On the other hand, the same phenomenon may be described in a more or less detailed way, thus leading to alternative descriptions differing in their grain size, which may be advantageous in special circumstances. The presence of alternative descriptions may, in fact, be helpful in an application: for instance, in machine translation, it may be desirable to have alternative descriptions of text meaning, because one of them may be easier for the generator to use in synthesizing the target text. From the point of view of a natural language sentence, the fact that it can be represented as two different TMRs is ambiguity. As far as TMRs are concerned, it is, of course, synonymy. As we will demonstrate in Section 6.6, the extant implementations of ontological semantics have never consciously allowed for TMR synonymy.

2.4.2 Parameters Related to the Methodology Associated with a Theory

2.4.2.1 Methodology and Linguistic Theory

Issues related to methodology in linguistic theory have been largely neglected, in part, due to Chomsky’s (1957: 50-53; 1965: 18-20) belief that no rigorous procedure of theory discovery was possible in principle and that methodological decisions involved in that activity were attained through trial and error and taking into account prior experience. What happens in the implementation of the linguistic theory methodologically apparently depends on its value on the parameter of effectiveness. In constructive theories, the methodological task is to see whether the ideal methodology which “comes with” a theory is executable directly or whether it should be replaced by a more efficient methodology. In linguistics, most constructive theories have relatively small purviews and simple bodies. A simplistic example, for illustration purposes only, would be a theory of feature composition (say, 24 features) for the phonemes (say, 50 in number) of a natural language.

Most linguistic theories, however, are non-constructive and often ineffective, that is, there is no obvious algorithm for their realization, that is, for generating descriptions associated with the theory. Typically, methodological activity in such theories involves the search for a single rule to account for a phenomenon under consideration. After such a rule, say, that for cliticization, is formulated on a limited material, for instance, one natural language, it is applied to a larger set of similar phenomena, for instance, the clitics in other natural languages. Eventually, the rule is modified, improved and accepted. Inevitably, in every known instance of this method at work, a hard residue of phenomena remains that cannot be accounted for by even the modified and

improved rule. More seriously, however, the work on the rule in question never concerns itself with connecting to rules describing adjacent phenomena, thus precluding any comprehensive description of language. This amounts to neoatomicism: one rule at a time instead of the pre-structuralist one phenomenon at a time. The expectation in such an approach is that all the other rules in language will fall in somehow with the one being described, an expectation never actually confirmed in an implementation. This is why in its own implementations ontological semantics develops microtheories, no matter how limited in purview, which are informed by the need to integrate them for the purpose of achieving a complete description.

In principle, linguistic theories profess to strive to produce complete descriptions of all the data in their purview. In practice, however, corners are cut—not that we are against or above cutting corners (e.g., under the banner of grain size); but they should be the appropriate corners, and they must not be too numerous. When faced with the abovementioned hard residue of data that does not lend itself to processing by the rule system proposed for the phenomenon in question, linguists typically use one of two general strategies. One is to focus on treating this hard residue at the expense of the “ordinary case.” (The latter is assumed, gratuitously, to have been described fully.)²⁰ The other strategy is to discard the hard residue: by either declaring it out of the purview of the theory or by treating the incompleteness of the set of theoretical rules as methodologically acceptable. This latter option results in the ubiquity of etceteras at the end of rule sets or even lists of values of individual phenomena in many linguistic descriptions.

Our experience has shown that focusing on borderline and exceptional cases often leaves the ordinary case underdescribed. Thus, for instance, in the literature on adjectival semantics, much attention has been paid to the phenomenon of relative adjectives developing a secondary qualitative meaning (e.g., *wooden* (table) > *wooden* (smile)). The number of such shifts in any language is limited. At the same time, as shown in Raskin and Nirenburg (1995), the scalar adjectives, which constitute one of the largest classes of adjectives in any language, are not described in literature much beyond an occasional statement that they are scalar.

Describing the ordinary case becomes less important when the preferred model of scientific progress in linguistics stresses incremental improvement by focusing on one rule at a time. Exceptions to rules can, of course, be simply enumerated, with their properties described separately for each case. This way of describing data is known as extensional definition. The complementary way of describing data through rules is known as intensional. Intensional definitions are seen by theoretical linguists as more valuable because they promise to cover several phenomena in one go. In discussing the relations between theories, methodologies and applications, we will show that the best methodology for a practical application should judiciously combine the intensional and extensional approach, so as to minimize resource expenditure (see Section 2.5 below).

20. This methodological bias is not limited to linguistics. It was for a very similar transgression that Bar Hillel criticized the methodology of logical semanticists: they unduly constrain their purview, and within that limited purview, concentrate primarily on exceptions: “One major prejudice... is the tendency to assign truth values to indicative sentences in natural languages and to look at those cases where such a procedure seems to be somehow wrong...” (Bar Hillel 1970: 203).

2.4.2.2 Methodology and AI

The somewhat shaky status of methodology in linguistic theory is an example of what can be termed a “subject-specific methodological problem” (Pandit 1991: 167-168). In AI, the other parent of NLP, we find modeling as the only methodological verity in the discipline. Under the “strong AI thesis” (we will use the formulation by the philosopher John Searle 1980: 353; see also Searle 1982a; cf. Searle 1997), “the appropriately programmed computer really *is* a mind, in the sense that computers given the right programs can be literally said to understand and have other cognitive states,” a claim that Searle ascribes to Turing (1950) and that forms the basis of the Turing Test. We agree with Moody (1993: 79), that “[i]t is an open question whether strong AI really does represent a commitment of most or many researchers in AI” (see also 2.4.3 below).

So instead of modeling the mind itself, under the “weak AI thesis” “the study of the mind can be advanced by developing and studying computer models of various mental processes” (Moody, 1993: 79-80). We part company with Moody, however, when he continues that “[a]lthough weak AI is of considerable methodological interest in cognitive science, it is not of much philosophical interest” (*op.cit.*: 80). The whole point of this chapter is to show how the philosophical, foundational approach to NLP, viewed as a form of weak AI, enhances and enriches its practice.²¹

2.4.2.3 Methodology and the Philosophy of Science

The philosophy of science does not have that much to say about the methodology of science. What is of general philosophical interest as far as methodological issues are concerned is the most abstract considerations about directions or goals of scientific research. Dilworth (1994: 50-51 and 68-70), for instance, shows how immediately and intricately methodology is connected to and determined by ontology: without understanding how things are in the field of research it is impossible to understand what to do in order to advance the field. At this abstract level, the questions that are addressed in the philosophy of science are, typically, the essentialist “‘*what*-questions’ and explanatory-seeking ‘*why*-questions’” (Pandit, 1991: 100), but not the *how*-questions that we will address in the next section and again in Section 2.5 below.

2.4.2.4 Methodology of Discovery: Heuristics

One crucial kind of *how*-questions, still of a rather abstract nature, has to do with discovery. In theoretical linguistics this may be posed as the problem of grammar discovery: given a set of grammatical data, e.g., a corpus, one sets out to discover a grammar that fits the data. Chomsky (1957) denies the possibility of achieving this goal formally. AI seems similarly sceptical about automatic discovery, not only of theory but even of heuristics: “[t]he history of Artificial Intelligence shows us that heuristics are difficult to delineate in a clear-cut manner and that the convergence of ideas about their nature is very slow” (Groner *et al.*, 1983b: 16).

Various described, as “rules of thumb and bits of knowledge, useful (though not guaranteed) for making various selections and evaluations” (Newell, 1983: 210), “strategic principles of demon-

21. We understand what Moody means by “philosophical interest,” however. On the one hand, it is the fascinating if still tentative philosophy of the mind (see, for instance, Simon 1979, 1989, Fodor 1990, 1994, Jackendoff 1994); on the other, it is the recurring fashion for imagination-stimulating, science fiction-inspired punditry in the media about robots and thinking machines and the philosophical ramifications of their future existence.

strated usefulness” (Moody, 1993: 105), or—more specifically—in knowledge engineering for expert systems (see, for instance, Mitchie 1979, Forsyth and Rada 1986, Durkin 1994, Stefik 1995, Awad 1996, Wagner 1998), “the informal judgmental rules that guide [the expert],” (Lenat 1983: 352), heuristics seem to be tools for the discovery of new knowledge. Over the centuries, they have been considered and presented as important road signs guiding human intelligence.

Heuristics as the art, or science, of discovery (and, therefore, used in the singular) is viewed as originating with Plato or even the Pythagoreans, who preceded him in the 6th century B.C.E. The field eventually concentrated on two major concepts, analysis and synthesis. The method of analysis prescribed the dissection of a problem, recursively, if necessary, into smaller and, eventually, familiar elements. Synthesis combined familiar elements to form a solution for a new problem. It is not so hard to recognize in these the contemporary top-down and bottom-up, or deductive and inductive, empirical approaches.

Later, heuristics was appropriated by mathematics and turned into a search for algorithms. Descartes (1908, see also Groner *et al.* 1983a and D. Attardo 1996) finalized this conception as 21 major heuristic rules applicable to problems presented algebraically. His more general heuristic recommendations call for a careful study of the problem until clear understanding is achieved, the use of the senses, memory, and imagination, and a great deal of practice, solving problems that have already been solved by others.

More recently, heuristics has been adopted by the philosophy of science and has become more openly subject-specific than its new parent discipline: there are the heuristics of physics (e.g., Bolzano 1930, Zwicky 1957, 1966, Bunge 1967, Post 1971), psychology (e.g., Mayer and Orth 1901, Bühler 1907, and Müller 1911, all of the Würzburg School, as well as Selz 1935 and, most influentially, Duncker 1935), and, of course, mathematics, where Descartes was revived and Polya’s work (1945, 1954a,b, 1962, 1965) became influential if not definitive.

Newell (1983) brought Polya to the attention of the AI community and suggested that AI should model the four major problem solving steps that Polya postulated—understanding the problem, devising a plan, carrying it out, and examining solutions (see Polya 1945 and Newell 1983: 203)—in automatic systems of discovery and learning.²² The heuristics of other disciplines look very much like Polya’s recommendations. They helpfully dissect a potentially complex problem into small steps. They all fall short of explaining specifically, other than with the help of examples, how the dissection should be implemented and how each step is to be performed. It was this aspect of heuristics that led Leibniz (1880) to criticizing Descartes and satirizing his rules that were too general to be useful: *Sume quod debes et operare ut debes, et habebis quod optas* (“Take

22. A considerable amount of interesting contributions in AI heuristics (see Zanakis *et al.* 1989 for an early survey) developed Newell and Simon’s general ideas on problem solving (Newell and Simon 1961, Newell *et al.* 1958, Newell and Simon 1972, Simon 1977, 1983), from automating discovery strategy in largely mathematical toy domains (e.g., Lenat 1982, 1983) to a densely populated area of heuristic search techniques (e.g., Lawler and Wood 1966, Nilsson 1971, Pearl 1984, Reeves 1993, Rayward-Smith 1995) to considerable initial progress in automatic theorem solving (see, for instance, Gelernter and Rochester 1958, Gelernter 1959, 1963, Gelernter *et al.* 1963) and machine learning (see, for instance, Forsyth and Rada 1986, Shavlik and Dietterich 1990, Kearns and Vazirani 1994, Langley 1996, Mitchell 1997).

what you have to take, and work the way you have to, and you will get what you are looking for”) (Vol. IV: 329; see also Groner *et al.* 1983b: 6).

2.4.2.5 Practical Skills and Tools as Part of Methodology

While the idea of heuristics has considerable appeal, we have to doubt its practical usefulness on at least two counts. First, our personal problem-solving experience seems to suggest that after the work is done it is not hard to identify, *post-hoc*, some of Polya’s steps in the way the solutions were reached.²³ In the process of solving the problems, however, we were not aware of these steps nor of following them. Nor, to be fair, were we aware of operating combinatorially Leibniz’s “alphabet of human thoughts,” the basis of his “generative lexicon” of all known and new ideas (see, for instance, his 1880, Vol. I: 57 as well as Groner *et al.* 1983b: 6-7). Nor did we count a great deal on insights, leading to a sudden and definitively helpful reorganization of a problem (cf. Köhler 1921, Wertheimer 1945).

We do see pedagogical value in Polya’s and others’ heuristics but we also realize, on the basis of our own experiences as students and teachers, that one cannot learn to do one’s trade by heuristics alone. If we look at the few examples of linguistic work on heuristics, we discover, along with attempts to apply general heuristics to the specific field of language (Botha 1981, Pericliev 1990), some useful heuristics for linguistic description (Crombie 1985, Mel’čuk 1988, Raskin and Nirenburg 1995, Viegas and Raskin 1998). However, we fully recognize how much should be learned about the field prior to studying and attempting to apply the heuristics. Similarly, in AI, one should learn programming and algorithm design before attempting to devise heuristics. All these basic skills are part of methodology, though they have often been taken for granted or even considered as pure engineering skills in the philosophical discussions of methodology.

These actual skills are responses to the unpopular *how*-questions that philosophy of science (or philosophy of language, for that matter, and philosophy in general) never actually asks. We agree with Leibniz’s critique of Descartes from this point of view too: heuristics are answers to *what*-questions, but how about *how*?

What does a computational linguist need to know to do his or her work? An indirect answer can be: what they are taught in school. In other words, if what linguists are taught prepares them for plying the trade, then the contents of the linguistics courses are the skills that linguists need. The actual truth is, of course, that linguists end up discarding or at least ignoring a part of what they are taught and supplementing their skills with those acquired on their own.

As we mentioned above, a typical contemporary linguistic enterprise involves a study of how a certain system of grammar fits a phenomenon in a natural language and how the grammar may

23. Just as it was easy to believe that we had gone through Dewey’s (1910: 74-104) five psychological phases of problem solving, viz., suggestion, intellectualization, the guiding idea, reasoning, and testing hypotheses by action, or Wallas’s (1926: 79-107) psychological steps, namely, preparation, incubation, illumination, and verification, or even psychotherapist Moustakas’s (1990) six phases of heuristic research: initial engagement, immersion, incubation, illumination, explication, creative synthesis. Somewhat more substantively, we definitely recognized various forms of guessing and judging under uncertainty, i.e., essentially engaging in certain forms of abduction, as explored by Tversky and Kahneman (1973—see also Kahneman *et al.* 1982; cf. Heath and Tindale 1994).

have to be modified to achieve a better match. This can vary to include a class of phenomena, a set of languages, or sometimes a comparison of two competing grammars. Somewhat simplistically, we can view such a linguistic task as requiring a grammatical paradigm, for instance, lexical-functional grammar, along with all the knowledge necessary for the complete understanding of the paradigm by the linguist, a native speaker of a language (or, alternatively, a representative corpus for the language), and algorithms for recognizing language phenomena as members of certain grammatical and lexical categories and of classes described by certain rules established by the paradigm.

On this view, the linguist starts with an empty template, as it were, provided by a grammatical system and finishes when the template is filled out by the material of the language described. Practically, of course, the research always deals with a limited set of phenomena, and then with specific features of that set. This limitation leads to the development of microtheories, in our terminology (see Section 2.4.4. below).

Similarly, an AI expert needs specific skills that he or she acquires in the process of training in computer science and/or directly in AI. This includes basic programming skills, familiarity with a number of programming languages, and modeling skills, involving the ability to build an architecture for an AI solution to a problem and knowledge of a large library of standard computer routines.

A complete methodology, then, includes both higher-level, at least partially heuristics-based ways of dissecting a new problem and lower-level disciplinary skills, sometimes—and certainly in the case of NLP—from more than one discipline. How does such a complete methodology interact with theory?

2.4.2.6 Disequilibrium Between Theory and Methodology

Within an established, ideal paradigm, one expects an equilibrium between the theory and methodology. The latter is also expected to determine the kind of descriptions that are needed to solve the problems and to achieve the goals of the field within that paradigm. Because no active discipline is complete and fully implemented, there is a continuous tug of war, as it were, between the theory of a field and its methodology: as more and more descriptions become necessary, the methodology must develop new tools to implement the expanded goals; as the implementation potential of the methodology grows it may lead to the implementation of new descriptions, and the theory may need to be expanded or modified to accommodate these gains.

In this creative disequilibrium, if the methodology, especially one based on a single method, is allowed to define the purview of a field, we end up with a ubiquitous method-driven approach. Chomskian linguistics is the most prominent example in linguistics, actively defining anything it cannot handle out of the field and having to revise the disciplinary boundaries for internal reasons, as its toolbox expands, and for external reasons, when it tries to incorporate the areas previously untouched by it or developed within a rival paradigm. The problem-driven approach, on the other hand, rejects the neatness of a single method on the grounds of principled unattainability. Instead, it must plunge headlong into the scruffiness of a realistic problem-solving situation, which always requires an ever-developing and expanding methodology, leading to inevitably eclectic, hybrid toolboxes.

2.4.2.7 Specific Methodology-Related Parameters

Several specific parameters follow from the discussion in Sections 2.4.2.1-6 above. The tension between theory building and goal-oriented description of phenomena creates the major parameter in this class, that of **method-driven** (“supply-side”) vs. **problem-driven** (“demand-side”) approaches (see Nirenburg and Raskin 1996, 1999). Other parameters follow more or less obviously from the discussion in this section. If a theory is effective in the sense of 2.4.1.2, it “comes” with a methodology but the methodology may be not **machine-tractable**. Whether it is or not, constitutes another methodology-related parameter, this one limited to effective theories. A theory may come packaged with a set of clear subject-specific heuristics, and if it does, this is a value of yet another parameter, **heuristics availability**. A similarly formulated parameter concerns the availability of a clear set of **skills/tools** associated with the purview of the theory.

2.4.3 Parameters Related to the Status of Theory as Model of Human Behavior

A formal or computational theory may or may not make a claim that it is a model of a natural process. The most well-known claim of this sort is the “strong AI hypothesis” (see also Section 2.4.2.2) which sees AI “as relevant to psychology, insofar as [it takes] a computational approach to psychological phenomena. The essence of the computational viewpoint is that at least some, and perhaps all, aspects of the mind can be fruitfully described for theoretical purposes by using computational concepts” (Boden 1981: 71-72). Whether a theory makes **strong hypothesis** claims constitutes a parameter.

This issue is actually an instance of the central question of the philosophy of science, namely, the status of theoretical categories and constructs with regard to reality, which we already touched upon in the discussion of justification in Section 2.3.4. While going over the extensive discussions of this issue in philosophical literature, we could not help wondering why we could not strongly identify our own theory of ontological semantics with any one of the rival positions. The most appealing position seems to be the least extreme one. A version of realism, it assumes a coexistence within the same theory of categories and constructs which exist in reality with those that are products of the mind, as long as the statements about both kinds are coherent with each other.

We finally realized that the reason for our lack of strong identification, as well as a half-hearted commitment to one of the positions, is due to the fact that ontological semantics does not aspire to the status of a strong hypothesis. In other words, it does not claim any psychological reality. It does not claim that humans store word senses, concepts, or sentential meaning in the format developed in ontological semantics for the lexicon, ontology or TMRs, respectively. Nor does this claim extend to equating in any way the processes of human understanding or production of sentences with the mechanisms for analysis and synthesis of texts in ontological semantics. We do not think that this takes away from the status of ontological semantics in the realm of science.

2.4.4 Parameters Related to the Internal Organization of a Theory

When dealing with a purview of considerable size, the pure theorist may be driven away from the natural desire to put forward a single comprehensive theory by the sheer complexity of the task. The alternative strategy is to break the purview up into chunks, develop separate theories for each of them and then to integrate them. This has been common practice in linguistics as well as in other disciplines, though the integration task received relatively little attention. Ontological

semantics has undergone such chunking, too. In it, we call the components of the theory microtheories (see Section 1.7). The microtheories can be circumscribed on the basis of a variety of approaches. There are microtheories devoted to language in general or particular languages; to different lexical categories, syntactic constructions, semantic and pragmatic phenomena or any other linguistic category; to world knowledge (“ontological”) phenomena underlying semantic descriptions; and to any of the processes involved in analysis and generation of texts by computer.

2.4.5 Parameter Values and Some Theories

We believe that it would be useful to characterize and compare computational linguistic theories in terms of the parameters suggested above. As we are not writing a handbook of the field, we will not discuss every known approach. That could have led to misunderstanding due to incompleteness of information, and—most seriously, as we indicated in Section 2.4.1.3 above—the lack of theoretical explicitness of many approaches.²⁴ Besides, the parameters we suggested are not binary: rather, their multiple values seem to reflect a pretty complex “modal logic.” An admittedly incomplete survey of the field of linguistic and computational semantics (see also Nirenburg and Raskin 1996), has yielded the parameter values listed as row headers in Table 1. The columns of the table correspond to the tests for determining what value of a given parameter is assigned in a theory. In order to determine what value a parameter is to be assigned in Theory X we should go, for each such candidate parameter, through the following test consisting of seven steps inquiring if:

- the theory overtly addresses the parameter,
- the theory develops it,
- addressing the parameter falls within the purview of the theory,
- the parameter is possible in the theory,
- the parameter is necessary for it,
- the parameter is at all compatible with the theory,
- the status of the parameter in the theory is at all determinable.

For each parameter, the outcome of this test is a seven-element set of answers that together determine the value of this parameter. Each combination of answers is assigned a name. For example, the set “yes, yes, yes, yes, yes/no, yes, yes” is called DD, that is, this parameter is considered “declared and developed” in the theory. The names are used only as mnemonic devices. The interpretation of the actual labels is not important. What counts is the actual differences in the answer sets. The *yes/no* answer means that this test is not relevant for a given parameter value. Each named set of answers forms a row in Table 1.

In almost direct contradiction to the bold statement in Footnote 24, we proceed to illustrate in Table 2, somewhat irresponsibly and as non-judgmentally as possible, how the parameters introduced in this section apply to four sample theories, Bloomfield’s (1933) descriptive (structuralist) linguistics, Chomsky’s (1965) Standard Theory, Pustejovsky’s (1995) Generative Lexicon, and

24. In other words, we decline to follow the path, memorably marked by Lakoff (1971), when, in the opening salvos of the official warfare in early transformational semantics, he projected what his foes in interpretive semantics would do if they made a step they had not made. and proceeded to attack them for that hypothetically ascribed stance.

ontological semantics. In doing so, we ignore yet another complication in assigning parameters to theories: that judgments about parameter values are often impossible to make with respect to an entire theory—parameter values may refer only to some component of a theory and be undefined or difficult to interpret for other components.

Table 1: Types of Values for A Parameter

Parameter Value Test \ Parameter Value Name	Declared by Theory?	Developed in Theory?	Within Purview of Theory?	Possible in Theory?	Necessary for Theory?	Compatible with Theory?	Determinable in Theory?
Declared, developed (DD)	yes	yes	yes	yes	yes/no	yes	yes
Declared, part-developed (DP)	yes	partially	yes	yes	yes/no	yes	yes
Declared, possible (DO)	yes	no	yes	yes	yes/no	yes	yes
Declared, Non-Purview (DU)	yes	no	no	yes	no	yes/no	yes
Declared, Purview (DR)	yes	no	yes	no	no	yes	yes
Impossible (IM)	yes/no	no	no	no	no	no	yes
Undeclared, Possible, Unnecessary (UU)	no	no	yes/no	yes	no	yes	yes
Undeclared, Necessary (UN)	no	no	yes	yes	yes	yes	yes

Table 1: Types of Values for A Parameter

Parameter Value Test \ Parameter Value Name	Declared by Theory?	Developed in Theory?	Within Purview of Theory?	Possible in Theory?	Necessary for Theory?	Compatible with Theory?	Determinable in Theory?
Undeclared, Part-Developed (UP)	no	partially	yes	yes	yes/no	yes	yes
Indeterminable (IN)	yes/no	yes/no	yes/no	yes/no	yes/no	yes/no	no

What makes the parametrization of a theory complex is that the status of a theory with regard to each parameter may vary. The tests, in addition, are not necessarily independent of each other. Besides, the same parameter value named in the first column may correspond to several combinations of results of the parameter tests: thus, because of all those “yes/no” values in the last, the value of a parameter in a theory may be “Undeterminable (IN)” for 2^6 combinations of test result situations of the parameter assigned that value in a theory.

The 11 parameters in Table 2 are the ones listed and described in Sections 2.4.1.4 above, namely, adequacy (Ad), effectiveness (Ef), explicitness (Ex), formality (Fy), formalism (Fm), ambiguity (Am), method-drivenness (as opposed to problem-drivenness) (Md), machine tractability (Mt), heuristics availability (Ha), strong hypothesis (as in strong AI) (Sh), and internal organization as microtheories (Mi).

Table 2: Illustration of Parameter Values and Sample Theories

Parameter \ Theory	Ad	Ef	Ex	Fy	Fm	Am	Md	Mt	Ha	Sh	Mi
Descr. Ling	UN	UN	IM	DD	UU	IN	DD	IM	IM	IM	UU
St. Theory	DP	DD	DP	DD	DD	UN	DD	IM	IM	DO	UU
Gen. Lex.	UN	UN	UN	DD	DP	IN	DD	IN	IN	UU	UU
Ont. Sem	DP	DD	DD	DD	UU	DP	IM	DD	DP	UU	DP

Table 2 claims then, for instance, that the Generative Lexicon theory does not address such parameters as adequacy, efficiency, and explicitness; it declares and develops formality and

method-drivenness; it addresses and partially develops its formalism; it does not address its status with regard to the strong hypothesis and internal organization, the two unnecessary but possible parameters in this theory; and there is no information to help to determine its take on the theoretical parameters of ambiguity, machine tractability, and the availability of heuristics.

Ontological semantics, by contrast, addresses and develops effectiveness, explicitness, formality, and machine tractability; it addresses and partially develops adequacy, ambiguity and availability of heuristics; it does not address such possible but unnecessary parameters as formalism and strong hypothesis while method-drivenness is excluded.

In Section 2.6 below, a more responsible and detailed illustration of the values of just one parameter, explicitness, will be presented on the material of ontological semantics, the one theory we can vouch for with some confidence.

2.5 Relations Among Theory, Methodology and Applications

In the sections above, we have discussed theories, their components and their relations to methodology and description. In this section, we venture into the connections of theories with their applications.

2.5.1 Theories and Applications

Theories can be pursued for the sake of pure knowledge. Some theories can also be used in applications—in other words, they are applicable (or applied) theories. Applications are tasks whose main purpose is different from acquiring knowledge about the world of phenomena. Rather, applications usually have to do with tasks directed at creating new tools or other artifacts. We have preached (Nirenburg and Raskin 1987a,b, Raskin 1987a,b) and practiced (Raskin and Nirenburg 1995, 1996a,b, Nirenburg and Raskin 1996, 1999) selective incorporation of components of linguistic theory into applied theories for natural language processing applications. Linguistic theories may contain categories, constructs and descriptions useful for concrete applications in full or at least in part. At the very least, reference to the sum of linguistic knowledge may help NLP practitioners to avoid reinventing various wheels. The relations among theories, applications and methodologies are summarized in Figures 13-16. The findings hold not only for linguistic theories but for theories in general.

In addition to describing natural phenomena, people create artifacts.

Some such artifacts ("application systems") are tools for production of other artifacts ("application results").

Artifacts may become objects of study of theories (for example, all mathematical objects are artifacts!).



Figure 13. Applications and some of their characteristics.

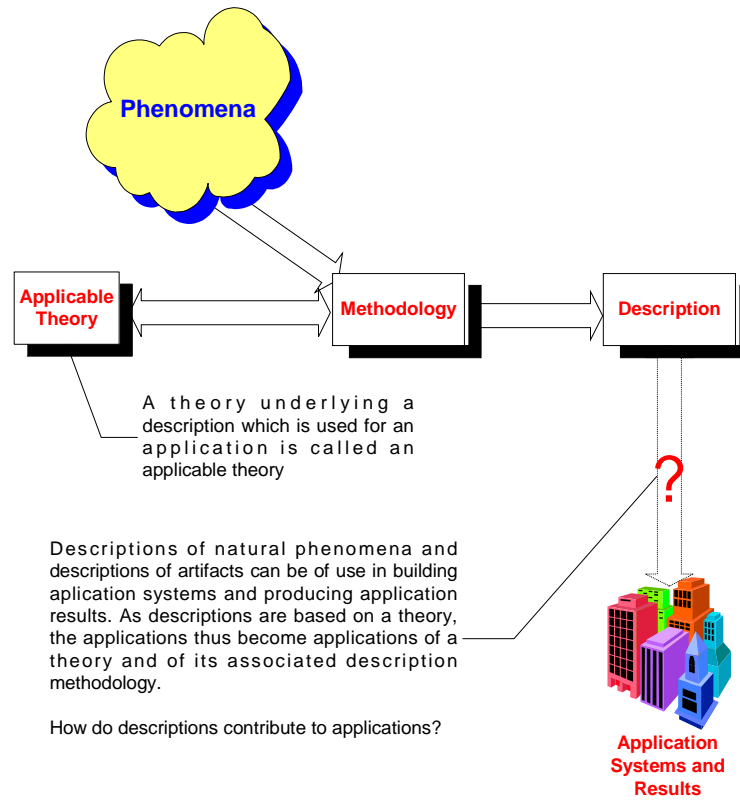


Figure 14. Introducing a different type of methodology.

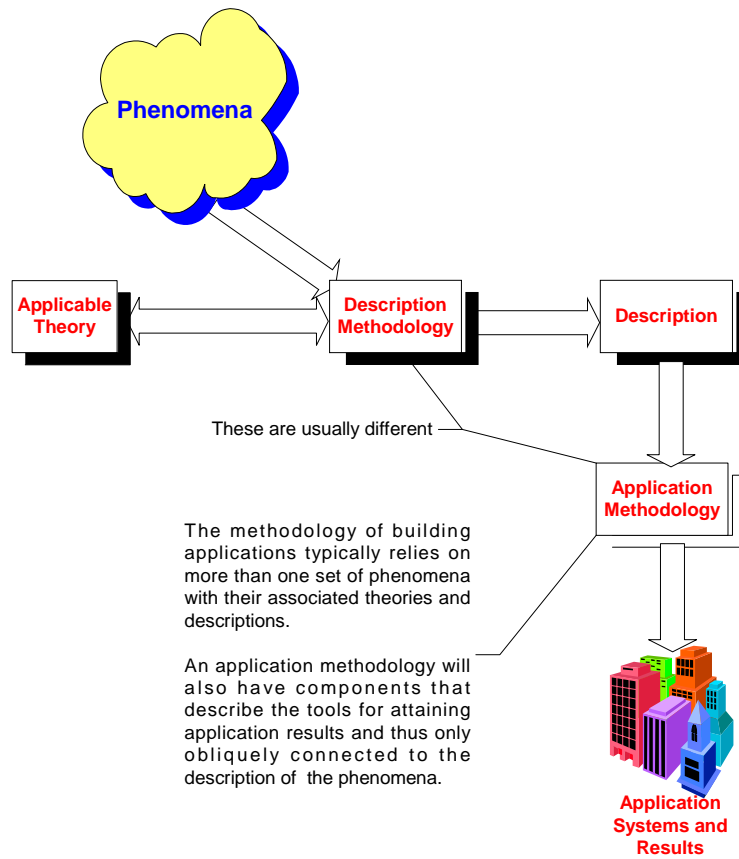


Figure 15. Some properties of application methodology

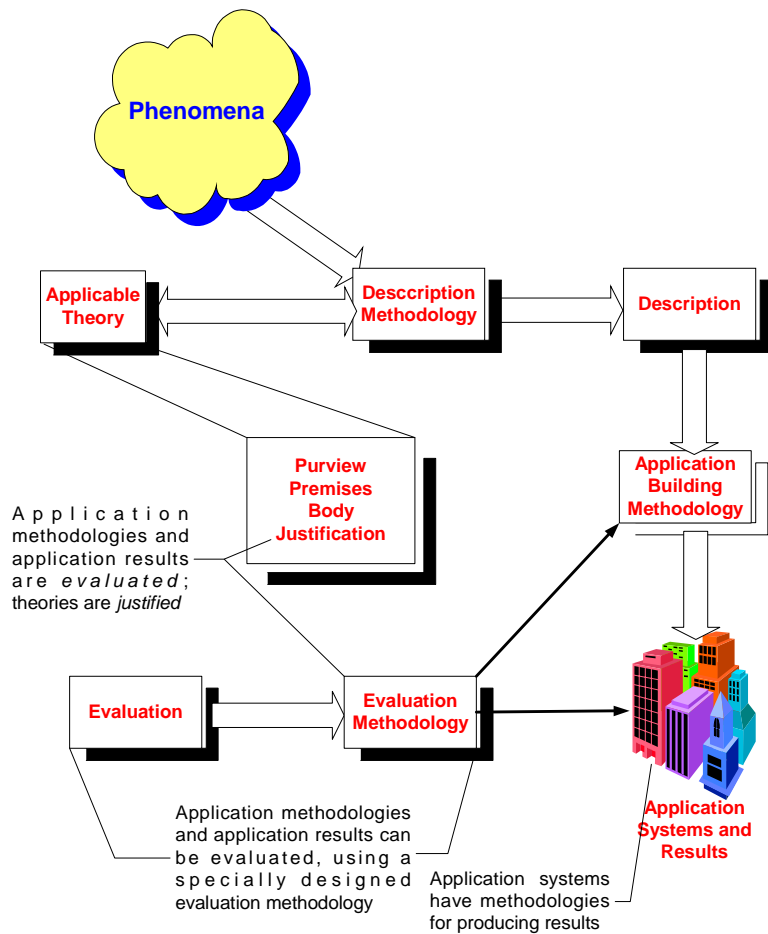


Figure 16. More types of methodologies: evaluation methodology and methodology of running applications, as opposed to the methodology of building applications.

There are, however, significant differences between applications and theoretical descriptions.

2.5.1.1 Difference 1: Goals

The first difference is in the goals of these pursuits. A theoretical linguistic description aims at modeling human language competence. Developing, say, a grammar of Tagalog qualifies as this kind of pursuit. By contrast, developing a learner’s grammar or a textbook of Tagalog are typical applications. The practical grammar or a textbook may include material from the theoretical grammar for the task of teaching Tagalog as a foreign language. This utilitarian applicational task is different from the descriptive theoretical task. An application is a system (often, a computational system) developed to perform a specific constructive task, not to explain a slice of reality. As such, it is also an engineering notion.

From the methodological point of view, the work on theoretical descriptions does not have to be completed before work on applications based on them can start. The learner’s grammar may be shorter and cruder than a theoretical grammar and still succeed in its application. In practice, an

application may precede a theoretical description and even provide an impetus for it. In fact, the history of research and development in machine translation (an application field) and theoretical computational linguistics is a prime example of exactly this state of affairs, where necessity (as understood then) was the mother of invention (of computational linguistic theories).

2.5.1.2 Difference 2: Attitude to Resources

The second difference between theories and applications is in their relation to the issue of resource availability. A theory is free of resource considerations and implies unlimited resources (expense, time, space, anything). In fact, implementing a linguistic theory can very well be considered an infinite task. Indeed, linguists have worked for several centuries describing various language issues but still have not come up with a complete and exhaustive description of any language or dialect, down to every detail of reasonable granularity. There are always things remaining to be concretized or researched. Complete description remains, however, a declared goal of science. Infinite pursuit of a complete theory seems to be a right guaranteed by a Ph.D. diploma, just as pursuit of happiness is an inalienable right guaranteed by the US Constitution.

In contrast to this, any high-quality application in linguistics requires a complete²⁵ description of the sublanguage necessary for attaining this application's purpose (e.g., a Russian-English MT system for texts in the field of atomic energy). By introducing resource-driven constraints, an application turns itself into a finite problem. A corresponding change in the methodology of research must ensue: concrete application-oriented methodologies crucially depend on resource considerations. Thus, in a computational application, the machine tractability of a description, totally absent in theoretical linguistics (see Footnote 17 above), becomes crucial. The above implies that methodologies for theoretical descriptions are usually different from application-oriented methodologies.

2.5.1.3 Difference 3: Evaluation

Yet another difference is that theories must be **justified** in the sense described above, while applications must be **evaluated** by comparing their results with human performance on the same task or, alternatively, with results produced by other applications. This means, for instance, that a particular learner's grammar of Tagalog can be evaluated as being better than another, say, by comparing examination grades of two groups of people who used the different grammars in their studies. No comparable measure can be put forward for a theoretical description.

2.5.2 Blame Assignment

An interesting aspect of evaluation is the difficult problem of "blame assignment": when the system works less than perfectly, it becomes desirable to pinpoint which component or components of the system is to blame for the substandard performance. Knowing how to assign blame is one of the most important diagnostic tools in system debugging. As this task is very hard, the real reasons why certain complex computational applications actually work or do not work are difficult to establish. As a result, many claims concerning the basis of a particular application in a particular theory cannot be readily proved. It is this state of affairs that led Wilks to formulate (only partially

25. Completeness is understood here relative to a certain given grain size of description. Without this *a priori* threshold, such descriptions may well be infinite.

in jest) the following “principle”: “There is no theory of language structure so ill-founded that it cannot be the basis for some successful MT” (Wilks 1992: 279). To extend this principle, even a theory which is seriously flawed, a theory which is not consistent or justifiable, infeasible and ineffective, can still contribute positively to an application.

The situation is further complicated by the fact that applications are rarely based exclusively on a single linguistic theory that was initially used as its basis. The modifications made in the process of building an application may, as we mentioned before, significantly change the nature of the theory components and parameters. Elements of other theories may find their way into an implementation. And finally, important decisions may be made by the developers which are not based on any overtly stated theory at all.²⁶

2.5.3 Methodologies for Applications

2.5.3.1 “Purity” of Methodology

An important methodological distinction between theories and applications has to do with the debate between method-oriented and problem-oriented approaches to scientific research (cf. Section 2.4.2.7 above, Nirenburg and Raskin 1999, Lehnert 1994). While it is tenable to pursue both approaches in working on a theory, applications, simply by their nature, instill the primacy of problem-orientedness. Every “pure” method is limited in its applicability, and in the general case, its purview may not completely cover the needs of an application. Fidelity to empirical evidence and simplicity and consistency of logical formulation are usually taken as the most general desiderata of scientific method, fidelity to the evidence taking precedence in cases of conflict (cf. Caws 1967: 339). An extension of these desiderata into the realm of application may result in the following methodological principle: satisfaction of the needs of the task and simplicity and consistency of the mechanism for its attainment are the most general desiderata of applied scientific work, with the satisfaction of the needs of the task taking precedence in cases of conflict.

2.5.3.2 Solutions are a Must, Even for Unsolvable Problems

In many cases, application tasks in NLP do not have proven methods that lead to their successful implementation. Arguably, some applications include tasks that are not solvable in principle. A well-known example of reasoning along these lines is Quine’s demonstration of the impossibility of translation between natural languages (1960). Quine introduces a situation in which a linguist and an informant work on the latter’s native language when a rabbit runs in front of them. The informant points to the rabbit and says *gavagai*. Quine’s contention is that there is no way for the linguist to know that this locution should be translated into English as “rabbit” or “inalienable rabbit part” or “rabbiting.” For a translation theorist, the acceptance of Quine’s view may mean giving up on a theory. A machine translation application will not be affected by this contention in any way. Quine and the linguistic theorist do not face the practical need to build a translation system; an MT application does. It must produce a translation, that is, find a working method, even in

26. One can adopt a view that any application is based on a theory, in a trivial sense, namely the theory that underlies it. In NLP practice, such a theory is not usually cognized by the developers, but the point we are making is that that theory will not typically coincide with any single linguistic theory. It will, in the general case, be a hybrid of elements of several theories and a smattering of elements not supported by a theory.

the absence of theoretical input.

A reasonable interpretation of Quine's claim is that no translation is possible without some loss of meaning. The truth of this tenet is something that every practical translator already knows from experience, and a number of devices have been used by human and even some machine translators to deal with this eventuality. Ample criticism has been leveled at Quine for this claim from a variety of quarters (Katz 1978: 209—see also Katz 1972/1974: 18-24, Frege 1963: 1, Tarski 1956: 19-21, Searle 1969: 19-21, Nirenburg and Goodman 1990). A recent proposal in the philosophy of science can be used to reject Quine's claim on purely philosophical grounds. It states that "what makes a particular activity scientific is not that the reality it uncovers meets the ideal, but that its deviation from the ideal is always something to be accounted for." (Dilworth 1996:4). In other words, unattainability of a theoretical ideal means not that the theory should be given up but that it should be supplemented by statements explaining the deviations of reality from the ideal. If this is true of theoretical pursuits, it is *a fortiori* so for applications.

2.5.4 Aspects of Interactions Among Theories, Applications, and Methodologies

2.5.4.1 Explicit Theory Building

How do theory, methodology and applications actually interact? One way of thinking about this is to observe the way computational linguists carry out their work in constructing theories, methodologies and applications. This is a difficult task because, in writing about their work, people understandably prefer to concentrate on results, not on the process of their own thinking.²⁷ Katz and Fodor (1963) provide one memorable example of building a theory by overtly stating the reasoning about how to carve out the purview of the theory to exclude the meaning of the sentence in context.²⁸ But they are in a pronounced minority. One reason for that, both in linguistics and in other disciplines, is a pretty standard division of labor between philosophers and scientists: the former are concerned about the foundational aspects of the disciplines and do not do primary research; the latter build and modify theories and do not deal with foundational issues.²⁹

2.5.4.2 Partial Interactions

When one analyzes the influence of theory on methodology and applications, it quickly becomes clear that often it is not an entire theory but only some of its components that have a direct impact on a methodology or an application. Some methods, like for instance, the well-established ones of

27. A series of interesting but largely inconclusive experiments was conducted within the protocol approach to invention in rhetoric and composition in the 1980s (see, for instance, Flower 1981 and references there; cf. Flower 1994). Writers were asked to comment on their thinking processes as they were composing a new text. On the use of the technique in cognitive science, see Ericsson and Simon (1993).

28. An even more ostentatious attempt in overt theory building is Katz and Postal (1964), where semantic reality was manipulated to fit into an imported premise, later abandoned in Revised Standard Theory (Chomsky 1971) that transformations did not change meaning.

29. As Moody (1993: 3) puts it, "[i]f the sciences are indeed the paradigms of progress, they achieve this status by somehow bypassing the foundational questions or, as was said earlier, by taking certain foundations for granted.... The practical rule in the sciences seems to be: Avoid confronting foundational questions until the avoidance blocks further progress." As this chapter documents, we do believe, on the basis of our practical experience, that we are at the stage in the development of NLP, computational semantics, and perhaps linguistic semantics in general, where "the avoidance blocks further progress."

field linguistics (see, for instance, Samarin 1967, Bouquiaux and Thomas 1992, Payne 1997), rely essentially on some premises of a theory (e.g., “the informant’s response to questions of the prescribed type is ultimate”) but not really on any universally accepted body: different field linguists will have different approaches to, for instance, syntax or morphology, which would be reflected in differences in the questions to the informant but not necessarily in the differences among the discovered phenomena.

2.5.4.3 Theoretical Premises Pertaining to Applications

One premise of computational linguistic theory pertaining directly to methodology of application building is that whenever successful and efficient **automatic** methods can be developed for a task, they are preferred over those involving humans. Another premise, which is probably quite universal among the sciences, is that if a **single method** can do the task, it is preferable to a combination of methods because combining methods can usually be done only at the cost of modifying them in some way to make them coexist. This premise is in opposition to yet another one: that recognizing theoretical overlaps between a new task and a previously accomplished one can save resources because some methods can be **reused**.

Yet another premise states that the need to create a **successful application** is more basic than the desire to do it using a single, automatic, logically consistent, and economical method. This tenet forces application builders to use a mixture of different techniques when a single technique does not deliver. But, additionally, when gaps remain, for which no adequate method can be developed, this tenet may lead application builders to using non-automatic methods as a way of guaranteeing success of an application. In practice, at the commercial end of the spectrum of comprehensive computational linguistic applications, a combination of human and automatic methods is a rule rather than an exception as is witnessed in many systems of human-aided machine translation, “workstations” for a variety of human analysts, etc.

Finally, there is the **resource** premise: applications must be built within the available resources of time and human effort and can only be considered successful if producing results in these application is also cost-effective in terms of resource expenditure. This premise is quite central for all applications, while in purely theoretical work it is of marginal importance. This is where it becomes clear that the theory underlying an application may vary from a theory underlying regular academic research motivated only by the desire to discover how things are.

2.5.4.4 Constraints on Automation

It is often resource-related concerns that bring human resources into an otherwise automatic system. Why specifically may human help within a system be necessary? Given an input, a computational linguistic application engine would produce application results algorithmically, that is, at each of a finite number of steps in the process, the system will know what to do and what to do next. If these decisions are made with less than complete certainty, the process becomes heuristic (see also Section 2.4.2.4 above). Heuristics are by definition defeasible. Moreover, in text processing, some inputs will always be unexpected, that is, such that solutions for some phenomena contained in them have not been thought through beforehand. This means that predetermined heuristics are bound to fail, in some cases. If this state of affairs is judged as unacceptable, then two options present themselves to the application builders: to use an expandable set of dynamically

modifiable heuristics to suit an unexpected situation (as most artificial intelligence programs would like to but typically are still unable to do) or to resort to a human “oracle.”

2.5.4.5 Real-Life Interactions

Irrespective of whether applications drive theories or theories license applications, it is fair to suppose that all research work starts with the specification of a task (of course, a task might be to investigate the properties of an application or a tool). The next thing may be to search for a methodology to carry this task out. This imported methodology may be general or specific, depending on the task and on the availability of a method developed for a different task but looking promising for the one at hand. A converse strategy is to start with developing an application methodology and then look for an application for it. An optional interim step here may be building a theory prior to looking for applications, but normally, the theory emerges immediately as the format of descriptions/results produced by the methodology.

2.5.5 Examples of Interactions Among Theories, Applications, and Methodologies

Of course, this discussion may be considered somewhat too general and belaboring the obvious, even if one goes into further detail on the types of theories, methodologies and applications that can interact in various ways. However, several examples can help to clarify the issues.

2.5.5.1 Statistics-Based Machine Translation

Let us start, briefly, with statistics-based machine translation. The name of this area of computational-linguistic study is a convenient combination of the name of an application with the name of a method. The best-developed effort in this area is the MT system *Candide*, developed at IBM Yorktown Heights Research Center (Brown *et al.* 1990).

It is not clear whether the impetus to its development was the desire to use a particular set of methods—already well established in speech processing by the time work on *Candide* started—for a new application, MT, or whether the methods were selected after the task was posited. The important point is that from the outset, *Candide* imported a method into a new domain. The statistical methods used in *Candide* (the trigram modeling of language; the source-target alignment algorithms, the Bayesian inference mechanism, etc.) were complemented by a specially developed theory. The theory was of text translation, not of language as a whole, and it essentially provided methodological guidelines for *Candide*. It stated, roughly, that the probability of a target language string *T* being a translation of a source language string *S* is proportional to the product of a) the probability that *T* is a legal string in the target language and b) the probability that *S* is a translation of *T*. In such formulation, these statements belong to the body of the theory, premised on a statement that probability (and frequency of strings in a text) affect its translation.

The task has been methodologically subdivided into two, corresponding to establishing the probabilities on the right hand side of the theoretical equation. For each of these subtasks, a complete methodology was constructed. It relied, among other things, on the availability of a very large bilingual corpus. In fact, had such a corpus not been available, it should have been constructed for the statistical translation methodology to work. And this would have drawn additional resources, in this case, possibly, rendering the entire methodology inapplicable. As it happened, the methodology initially selected by *Candide* did not succeed in producing results of acceptable quality, due

to the complexity of estimating the various probabilities used in the system and the rather low accuracy of the statistical models of the target language and of the translation process. To improve the quality of the output, the Candide project modified its methodology by including versions of morphological and syntactic analysis and some other computational linguistic methods into the process. As a result, the quality of Candide output came closer to, though never quite equaled, that of the best rule-based MT systems. The hybridization of the approach has never been given any theoretical status. Simply, in addition to the statistical theory of MT, Candide now (consciously or not) employed the theory underlying the morphological and syntactic analyzers and their respective lexicons. The application-building methodology has been modified in order to better satisfy the needs of an application. Had the Candide effort continued beyond 1995, it might have changed its methodology even further, in hopes to satisfy these needs.

2.5.5.2 Quick Ramp-Up Machine Translation Developer System

As another example, let us briefly consider the case of the project Expedition, under development at NMSU CRL since late 1997. The project's stated objective is to build an environment (that is, a tool, or an implemented methodology) which will allow fast development, by a small team with no trained linguist on it, of moderate-level machine translation capabilities from any language into English. As a resource-saving measure, the system is encouraged to make use of any available tool and/or resource that may help in this task. As specified, this application is a metatool, a system to help build systems.

Once the objectives of the application have been stated, several methodologies could be suggested for it. These methodologies roughly fall into two broad classes—the essentially corpus-based and the essentially knowledge-based³⁰ ones. The reasoning favoring the corpus-based approach is as follows. As the identity of the source language is not known beforehand, and preparing for all possible source languages is well beyond any available resources, the easiest thing to do for a new source language is to collect a corpus of texts in it and apply to it the statistical tools that are becoming increasingly standard in the field of computational linguistics: text segmentors for languages that do not use breaks between words, part of speech taggers, grammar induction algorithms, word sense disambiguation algorithms, etc. If a sizeable parallel corpus of the source language and English can be obtained, then a statistics-based machine translation engine could be imported and used in the project. However, the corpus-based work, when carried out with purity of method, is usually not devoted to complete applications, while when it is (as in the case of Candide), it requires a large doze of “conventional” language descriptions and system components.

The reasoning favoring the knowledge-based approach is as follows. As the target language in the application is fixed, a considerable amount of work can be prepackaged: the target application can be supplied with the English text generator and the English side of the lexical and structural transfer rules for any target language. Additionally, both the algorithms and grammar and lexicon writing formats for the source language can be largely fixed beforehand. What remains is facilitating the acquisition of knowledge about the source language, its lexical stock, its grammar and its lexical and grammatical correspondences to English. This is not an inconsiderable task. The variety

30. The term “knowledge-based” is used here in a broad sense to mean “relying on overtly specified linguistic knowledge about a particular language,” and not in its narrow sense of “machine translation based on artificial intelligence methods.”

of means of realization for lexical and grammatical meanings in natural languages is notoriously broad. For many languages, published grammars and machine-readable monolingual and bilingual dictionaries exist, but their use in computational applications, as practice in computational linguistics has shown (see, for instance, Amsler 1984, Boguraev 1986, Evens 1989, Wilks *et al.* 1990, Guo 1995), requires special resource expenditure, not incomparable to that for building an NLP system.

Creating the knowledge for natural language processing applications has occupied computational linguists for several generations, and has proved to be quite an expensive undertaking, even when the knowledge acquirers are well trained in the formats and methods of description and equipped with the best corpus analysis and interface tools. Considering that the users of the knowledge elicitation tool will not be trained linguists and also taking into account that the time allotted for developing the underlying application (the machine translation system) is limited, the “traditional” approach to knowledge acquisition (notably, with the acquirer initiating all activity) has never been a viable option. The best methodological solution, under the circumstances, is to develop an interactive system which guides the acquirer through the acquisition steps—in fact, an automatic system for language knowledge elicitation of the field-linguistics type. The difficulties associated with this methodology centrally include its novelty (no linguistic knowledge acquisition environment of this kind has ever been attempted) and the practical impossibility of anticipating every phenomenon in every possible source language.

The field of computational linguistics as a whole has, for the past five years or so, devoted a significant amount of effort to finding ways for mixing corpus- and rule-based methods, in the spirit of the central methodological principle for building applications discussed above.³¹ The Expedition project is no exception. However, based on the expected availability of resources (the project’s main thrust is toward processing the less described, “low-density” languages) and on the generality of the task, the “classical” computational linguistic methodology was selected as the backbone of the project. A separate study has been launched into how to “import” any existing components for processing a source language into the Expedition system.

As no trained linguists will participate in the acquisition of knowledge about the source language, it was decided that, for pedagogical reasons, the work would proceed in two stages: first, the acquisition (elicitation) of a computationally relevant description of the language; then, the development of rules for processing inputs in that language, using processing modules which would be resident in the system. In both tasks, the system will hold much of the control initiative in the process. In order to do so, the elicitation system (in Expedition, it is called Boas, honoring Franz Boas (1858-1942), the founder of American descriptive linguistics, as well as a prominent anthropologist) must know what knowledge must be elicited. For our purposes in this chapter, a discussion of the first of the two tasks will suffice—see Nirenburg (1998b), Nirenburg and Raskin (1998) for a more detailed discussion of Boas.

31. The state of knowledge in this field is still pre-theoretical, as a variety of engineering solutions featuring eclectic methodology are propounded for a number of applications. It will be interesting to see whether a theory of merging tools and resources will gradually emerge. The work on computational-linguistic architectures (e.g., Cunningham *et al.* 1997a,b, Zajac *et al.* 1997) is, in fact, a step toward developing a format of a language to talk about such merges, which can be considered a part of the body of a theory.

The great variety of categories and expressive means in natural languages (as illustrated, for instance, by the complexity of tools and handbooks for field linguistics) is a major obstacle for Boas. *A priori*, the creation of a complete inventory of language features does not seem to be a realistic task. The goal of carrying it through to a concrete, systematic and applicable level of description is not attained—and not always attempted or even posited as an objective—by the workers in the areas of linguistic universals and universal language parameters (see, for instance, Greenberg 1978, Chomsky 1981, Berwick and Weinberg 1984, Webelhut 1992, Dorr 1993, Dorr *et al.* 1995, Kemenade and Vincent 1997). Methodologically, therefore, three choices exist for Boas: a data-driven method, a top-down, parameter-driven method, and some combination of these methods. As it happens, the last option is taken, just as in the case of the choice of corpus- or rule-based methodology for Expedition.

The data-driven, bottom-up strategy works in Boas, for example, for the acquisition of the source language lexicon, where a standard set of English word senses is given to the acquirer for translation into the source language.³² The top-down, parameter-oriented strategy works in elicitation of the morphological and syntactic categories of the language, together with their values and means of realization. Sometimes, these two strategies clash. For example, if closed-class lexical items, such as prepositions, are extracted in the lexicon, it is desirable (in fact, essential) for the purposes of further processing not only to establish their translations into English (or, in accordance with the Boas methodology, their source language translations, based on English) but also their semantics, in terms of what relation they realize (e.g., directional, temporal, possession, etc.). This is needed for disambiguating prepositions in translation (a notoriously difficult problem in standard syntax-oriented approaches to translation). In languages where the category of grammatical case is present, prepositions often “realize” the value of case jointly with case endings: for instance, in Russian, *s*+Genitive realizes a spatial relation of downward direction, with the emphasis on the origin of motion, as in “He jumped off the table”; *s*+Accusative realizes the comparative meaning: “It was as large as a house”; while *s*+Instrumental realizes the relation of “being together with”: “John and/together with Bill went to the movies” (see, for instance, Nirenburg 1980 for further discussion).

Under the given methodological division, however, Boas will acquire knowledge about case in the top-down, parameter-oriented way and information about prepositions in the bottom-up, data-driven way. For Russian, for instance, this knowledge will include the fact that the language features the parameter of case, that this parameter has six values and that these values are realized through inflectional morphology by suffixation, with the major inflectional paradigms listed. In order to reconcile the two approaches in this case, the lexical acquisition of prepositions for languages with grammatical case will have to include a question about what case form(s) a given preposition can introduce.

Note that throughout this discussion, a particular theory of language was assumed, as most of the categories, values, realizations and forms used in the descriptions have been introduced in a the-

32. It is possible to argue that single-meaning entries in the English vocabulary (not, of course, their combinations in complete entries for actual English words—cf. Nirenburg and Raskin 1998; Viegas and Raskin 1998) may serve as crude approximations for universal lexical-semantic parameters. Even on such an assumption, methodologically, the work of acquiring the source language lexicon remains very much empirical and data-driven.

ory, while work on Boas has essentially systematized and coordinated these theoretical concepts, adding new concepts mostly when needed for completeness of description. This is why, for example, the lists of potential values for the theoretical parameters adopted in Boas (such as case, number, syntactic agreement and others) are usually longer than those found in grammar books for individual languages and even general grammar books: the needs of the application necessitate extensions. Thus, for instance, the set of potential values of case in Boas includes more members than the list of sample cases in Blake (1994), even if one does not count name proliferation in different case systems for essentially the same cases.³³

For the purposes of this chapter, a central point of the above discussion is the analysis of the reasoning of the application builders. After some general methodological decisions were made, existing theories of language knowledge processing were consulted, namely the theories underlying the methodology of field linguistics and those underlying the study of universals; their utility and applicability to the task at hand were assessed and, as it happened, certain modifications were suggested in view of the peculiarities of the application. Of course, it was possible to “reinvent” these approaches to language description. However, the reliance on prior knowledge both saved time and gave the approaches used in the work on Boas a theoretical point of reference. Unfortunately, the actual descriptions produced by the above linguistic theories are of only oblique use in the application under discussion.

Boas itself is a nice example, on which one can see how theory, methodology, description, and application interact. The parameters for language description developed for Boas belong to the **body** of the theory underlying it. The general application-oriented methodological decisions (discussed above in terms of availability and nature of resources), together with the various specially developed front-end and back-end tools and procedures, constitute the **methodology** of Boas. The knowledge elicited from the user by Boas is the **description**. The resulting system is an **application**. Overt reasoning about methodology and theories helped in the formulation of Boas and Expedition. One can realistically expect that such reasoning will help other computational linguistic projects, too.

2.6 Using the Parameters

In this section, we discuss, by way of selective illustration, how the philosophical approach proposed here has been used to characterize and analyze ontological semantics. We concentrate on a single parameter: explicitness. Additionally, as of the four constituent parts (purview, premises, justification and body) of a theory, the body is, by nature, the most explicit (indeed, it is the only constituent described in most computational linguistic contributions), we will concentrate here on the other three constituents. A detailed statement about the body of ontological semantics is the subject of Part II of this book. Details about its various implementations have been published in and are cited abundantly throughout the book. To summarize, the analysis part of ontological semantic implementations interprets input sentences in a source language as meaning-rich text

33. Since several different theoretical traditions have been joined in Boas, to expand coverage, a methodological decision was made to include in the list of parameter values different aliases for the same value, to facilitate the work of the user by using terminology, to which he or she is habituated by the pertinent grammatical and/or pedagogical tradition.

meaning representations (TMRs), written in a metalanguage whose terms are based on an independent ontology. Word meanings are anchored in the ontology. The procedure of analysis, relies on the results of ecological, morphological and syntactic analysis and disambiguates and amalgamates the meanings of lexical items in the input into a TMR “formula.” The generation module takes the TMR, possibly augmented through reasoning over the ontology and the Fact DB, as input and produces natural language text for human consumption.

The main purpose for the discussion that follows is to articulate what it takes to go from relying on a covert, uncognized theory underlying any linguistic research, however application-oriented by design, to an inspectable, overt statement of the theoretical underpinnings of such an activity. This discussion is motivated and licensed by the conclusions about the benefits of using theory from Section 2.2.

2.6.1 Purview

The purview of ontological semantics is meaning in natural language. Meaning in ontological semantics can be static or dynamic. The former resides in lexical units (morphemes, words or phrasals) and is made explicit through their connections to ontological concepts. Dynamic meaning resides in representations of textual meaning (that is meaning of clauses, sentences, paragraphs and larger text units), produced and manipulated by the processing components of the theory. The theory, in effect, consists of a specification of how, for a given text, (static, context-independent) meanings of its elements (words, phrases, bound morphemes, word order, etc.) are combined into a (dynamic, context-dependent) text meaning representation, and vice versa. This is achieved with the help of static knowledge resources and processing components. The theory recognizes four types of static resources:

- an **ontology**, a language-independent compendium of information about the concepts underlying elements of natural language;
- a **fact database** (Fact DB), a language-independent repository of remembered instances of ontological concepts;
- a **lexicon**, containing information, expressed in terms of ontological concepts, about lexical items, both words phrasals; and
- an **onomasticon**, containing names and their acronyms.

The knowledge supporting the ecological, morphological and syntactic processing of texts is “external” to ontological semantics: much of the knowledge necessary for carrying out these three types of processing resides outside the static resources of ontological semantics—in morphological and syntactic grammars and ecological rule sets. However, some of this information actually finds its way in the ontological semantic lexicon, for example, to support linking.

The analyzer and the generator of text are the main text processing components. The reasoning module is the main application-oriented engine that manipulates TMRs. The term ‘dynamic,’ therefore, relates simply to the fact that there are no static repositories of contextual knowledge, and the processing modules are responsible for deriving meaning in context. A broader sense of dynamicity is that it serves the compositional property of language, having to do with combining meanings of text elements into the meaning of an entire text.

On the above view, the purview of ontological semantics includes that of formal semantics, which covers, in our terms, much of the grammatical meaning and parts of text meaning representation and adds the purview of lexical semantics. While the purview of ontological semantics is broader than that of formal semantics or lexical semantics, it is by no means unlimited. It does not, for instance, include, in the specification of the meaning of objects, any knowledge that is used by perception models for recognizing these objects in the real world.

2.6.2 Premises

In this section, we discuss several premises of ontological semantics and, whenever possible and to the best of our understanding, compare them with related premises of other theories. The premises we mention certainly do not form a complete set. Ontological semantics shares some premises with other scientific theories and many premises with other theories of language.

2.6.2.1 Premise 1: Meaning Should Be Studied and Represented

At the risk of sounding trivial or tautological, we will posit the first premise of ontological semantics as: “Meaning should be studied and represented.” This follows directly from the purview of our theory. We share the first part of the premise, that meaning should be studied, with all semanticists and philosophers of language and with knowledge-based strains in AI NLP but not with the linguists and computational linguists who constrain their interest to syntax or other areas.

We assume that meaning can and should be represented. We share this premise with most schools of thought in linguistics, AI and philosophy, with the notable exception of late Wittgenstein and the ordinary language philosophy (Wittgenstein 1953: I.10ff, especially, 40 and 43, Ryle 1949, 1953, Grice 1957, Austin 1961—see also Caton 1963, Chappell 1964) as well as some contributions within connectionism (see, for instance, Brooks 1991, Clark 1994), whose initial anti-representationalism has been on the retreat since Fodor and Pylyshyn’s (1988) challenge (e.g., Horgan and Tienson 1989, 1994, Pollack 1990, Berg 1992). Note that issues of the nature and the format of representation, such as levels of formality and/or machine tractability, belong in the body of the theory (see Footnote 17 and Section 2.4.1.4 above) and are, therefore, not discussed here.

2.6.2.2 Premise 2: The Need for Ontology

Ontological semantics does not have a strong stance concerning connections of meanings to the outside world (denotation, or extension relations). It certainly does not share the implicit verificationist premise of formal semanticists that the ability to determine the truth value of a statement expressed by a sentence equals the ability to understand the meaning of the sentence. One result of this difference is our lack of enthusiasm for truth values as semantic tools, at least for natural language, and especially as the exclusive tool of anchoring linguistic meanings in reality.

Unlike Wittgenstein and, following him, Wilks (e.g., 1972, 1982, 1992; Nirenburg and Wilks 1997), we still recognize as a premise of ontological semantics the existence of an (intensional) ontological signification level which defines not only the format but also the vocabulary (the metalanguage) of meaning description. While this level is distinct from denotation (it is not, directly, a part of the outside world), it is also distinct from language itself.

In ontological semantics, the English expressions *Morning star* and *Evening star* will both be

mapped into an instance of the ontological concept PLANET, namely, VENUS, which is stored in the fact database, while the corresponding English word *Venus* is listed in the English onomasticon (see 2.6.1 above). The Fact DB entry VENUS, in turn, is an instance of the ontological concept PLANET. It is this latter type of general concept that is the ultimate source of meaning for most individual open-class lexical units in ontological semantics.

Computational ontology, in its constructive, operational form as a knowledge base residing in computer memory, is not completely detached from the outside world, so that a variation of the familiar word-meaning-thing triangle (Ogden and Richards 1923; Stern 1931, Ullman 1951, Zvegintzev 1957), is still applicable here. The relation of the ontology to the outside world is imputed by the role ontological semantics assigns to human knowledge of the language and of the world—to interpret elements of the outside world and encode their properties in an ontology. As a corollary, the image of the outside world in ontological semantics includes entities which do not “exist” in the narrow sense of existence used in formal semantics; in this, we agree with Hirst (1991), where he follows Meinong (1904) and Parsons (1980).

For ontological semantics in action, the above triangular relation typically takes the form of sentence-meaning-event, where meaning is a statement in the text meaning representation (TMR) language and EVENT is an ontological concept. But ontological semantics is not completely solipsistic. The connection between the outside world (the realm of extension) and ontological semantics (the realm of intension) is carried out through the mediation of the human acquirer of the static knowledge sources.³⁴ This can be illustrated by the following example. The ontology contains a complex event MERGER, with two companies as its participants and a detailed list of component events, some of which are contingent on other components. In ontological semantics, this is a mental model of this complex event, specifically, a model of “how things can be in the world.” Ontological semantics in operation uses such mental models to generate concrete mental models about specific mergers, that is, “what actually happened” or even “what could happen” or “what did not happen.”

These latter models are not necessarily fleeting (even though a particular application of ontological semantics may not need such models once they are generated and used). In ontological semantics, they can be recorded as “remembered instances” in the knowledge base and used in subsequent NLP processing. Thus, for MERGER, remembered instances will include a description of the merger of Exxon and Mobil or of Chrysler and Daimler Benz. The remembered instances are intensional because they add a set of agentive, spatio-temporal and other “indices” to a complex event from the ontology.

We share with formal semanticists the concern for relating meaning to the outside world (cf. Lewis’s 1972 concern about “Markerese”) but we use a different tool for making this relation operational, namely, an ontology instead of truth values (see, for instance, Nirenburg *et al.* 1995). We basically accept the premises of mental model theorists (e.g., Johnson-Laird 1983, Fauconnier

34. When such acquisition is done semiautomatically, people still check the automatically produced results. If in the future a completely automatic procedure for knowledge acquisition is developed, this procedure will be recognized as a model of human intuition about the world and its connection to the ontology.

1985) that such models are necessary for semantic description, in particular, for accommodating entities that do not exist in the material sense. However, we take their concerns one step forward in that we actually construct the mental models in the ontology and the knowledge base of remembered instances.

We agree with the role Wittgenstein and Wilks assign to the real world, that is the lack of its direct involvement in the specification of meaning. We diverge from them in our preference for a meta-language that is distinct from natural language in the definitions of its lexical and syntactic units. Our position is explained by our desire to make meaning representation machine-tractable, that is, capable of being processed by computers. This desideratum does not obtain in the Wittgenstein/Wilks theoretical approach, whose motto, “meaning is other words,” seems, at least for practical purposes, to lead to a circularity, simply because natural language is notoriously difficult to process by computer, and this latter task is, in fact, the overall purpose and the starting point of the work in the field. Note that in his practical work, Wilks, a founder of computational semantics, does not, in fact, assume such a strong stance and does successfully use non-natural-language semantic representations (e.g., Wilks and Fass 1992).

This deserves, in fact, some further comment, underscoring the difference between Wilks, the application builder, and Wittgenstein (and possibly Wilks again), the theorist(s). The later Wittgenstein claim that “meaning is use” (see above) was non-representational: he and his followers made it clear that there could not exist an x , such that x is the meaning of some linguistic expression y . Wilks does say, throughout his work, that meaning is other words and thus sounds perfectly Wittgensteinian. In Wilks (1999), however, he finally clarifies an important point: for him, “other words” mean a complex representation of meaning—not a simple one-term-like entity of “uppercase” semantics. If this is the case, then not only is he Wittgensteinian, but so are Pustejovsky (1995) as well as ourselves—but Wittgenstein is not! Moreover, being ontology-oriented, which Wilks (1999) stops just barely short of, is then super-Wittgensteinian, as ontological semantics leads to even more intricate representations of meaning.

2.6.2.3 Premise 3: Machine Tractability

We are interested in machine-tractable representations of meaning (cf. Footnote 17 above) because of another premise, namely, that meaning can be manipulated by computer programs. We share this premise with many computational linguists and AI scholars but with few theoretical linguists or philosophers of language. For ontological semantics machine tractability goes hand in hand with the earlier premise of meaning representability. There are, however, some approaches that subscribe to the premise of machine tractability but not to the premise of meaning representability, e.g., the word sense disambiguation effort in corpus-oriented computational linguistics (e.g., Resnik and Yarowsky 1997, Yarowsky 1992, 1995, Cowie *et al.* 1992, Wilks *et al.* 1996, Wilks and Stevenson 1997; see, however, Kilgariff 1993, 1997a,b, and Wilks 1997).

2.6.2.4 Premise 4: Qualified Compositionality

Another important theoretical premise in the field is compositionality of meaning. It essentially states that the meaning of a whole is fully determined by the meanings of its parts and is usually applied to sentences as wholes and words as parts. Ontological semantics accepts this premise, but in a qualified way. The actual related premise in ontological semantics is as follows: while

sentence meaning is indeed largely determined by the meanings of words in the sentence, there are components of sentence meanings that cannot be traced back to an individual word and there are word meanings that do not individually contribute to sentence meaning. A trivial example of non-compositionality is the abundance of phrasal lexical units in any language. The main tradition in the philosophy of language (formal semantics) has, since Frege (1892), accepted complete compositionality as the central theoretical tenet. A variety of researchers have criticized this hypothesis as too strong for natural language (e.g., Wilks 1982). We concur with this criticism.

2.6.3 Justification

The justification component of ontological semantics is responsible for answering questions about why we do things the way we do. We see it as a process of reviewing the alternatives for a decision and making explicit the reasons for the choice of a particular purview, of premises and of the specific statements in the body.

While descriptive adequacy is a legitimate objective, and simplicity, elegance, and parsimony are generally accepted desiderata in any kind of scientific research, they are not defined specifically or constructively enough to be directly portable to ontological semantics. In any case, we are not sure to what extent the “Popperian justification tool” used in theoretical linguistics (see Section 2.3.4) is sufficient for ontological semantics or for the field of NLP in general. In fact, all debates in the NLP community about ways of building better NLP systems contribute to the justification of the (usually hidden) theories underlying the various methods and proposals—even when they are directly motivated by evaluations of applications.

Still, what is descriptive adequacy in ontological semantics? Surely, we want to describe our data as accurately as possible. To that end, it is customary in NLP to divide all the data into a training component and a test component, on which the description, carried out using the training component, is verified.

In principle, every statement in ontological semantics may be addressed from the point of view of justification. Thus, for example, in the Mikrokosmos implementation of ontological semantics, a choice had to be made between including information about lexical rule content and applicability in the lexicon or keeping it in a separate static knowledge source and using it at runtime (Viegas *et al.* 1996). The decision was made in favor of the former option because it was found experimentally that existence of exceptions to lexical rule applicability, which led some researchers to the study of a special device, “blocking,” to prevent incorrect application (see Ostler and Atkins 1991, Briscoe *et al.* 1995), made it preferable to mark each pertinent lexical entry explicitly as to whether a rule is applicable to it. Reasons for justifying a choice may include generality of coverage, economy of effort, expectation of better results, compatibility with other modules of the system and the theory and even availability of tools and resources, including availability of trained personnel.

The above example justifies a statement from the body of ontological semantics. We discover, however, that it is much more important and difficult to justify the purview and the premises of a theory than its body. Moreover, we maintain that the same premises can be combined with different bodies in the theory and still lead to the same results. The rule of thumb seems to be as follows: look how other NLP groups carry out a task, compare it with the way you go about it, and

find the essential differences. As we already mentioned, sociologically speaking, this job is the hardest within a large and homogeneous research community in which the examination of the theoretical underpinnings of the common activity may not be a condition of success. In what follows, an attempt is made to justify each of the stated premises of ontological semantics in turn.

2.6.3.1 Why should meaning be studied and represented?

We believe that meaning is needed, in the final analysis, to improve the output quality of NLP applications, in that it allows for better determination and disambiguation of structural, lexical and compositional properties of texts in a single language and across languages, and thus for better choices of target language elements in translation, or better fillers for information extraction templates, or better choices of components of texts for summarization. Knowledge of meaning presents grounds for preference among competing hypotheses at all levels of description, which can be seen especially clearly in a system, where evidence in the left hand side of rules can be of mixed—semantic, syntactic, etc.—provenance.

Reticence on the part of NLP workers towards meaning description is not uncommon and is based on the perception that the semantic work is either not well defined, or too complex, or too costly. Our practical experience seems to have demonstrated that it is possible to define this work in relatively simple terms; that it can be split into a small number of well-defined tasks (to be sure, a comprehensive treatment of a number of “hard residue” phenomena, such as metaphor, may still remain unsolved in an implementation, which is standard fare in all semantic analysis systems) and that, for the level of coverage attained, the resource expenditure is quite modest.

The above arguments are designed primarily for a debate with non-semantic-based rule-governed approaches (see, e.g., the brief descriptions in Chapters 10, 11, and 13-15 of Hutchins and Somers 1992). Now, from the standpoint of corpus-based NLP, the work of semantics can be done by establishing meaning relations without explaining them, directly, for example, on pairs of source and target language elements in MT. The task of integrating a set of target elements generated on the basis of a source language text through these uninterpreted correspondences into a coherent and meaningful target sentence becomes a separate task under this approach. It is also addressed in a purely statistics-based way by “smoothing” it, using comparisons with a target language model in the statistical sense (Brown and Frederking 1995).

2.6.3.2 Why is ontology needed?

It is practically and technologically impossible to operate with elements of the outside world as the realm of meaning for natural language elements. Therefore, if one wants to retain the capability of representing and manipulating meaning, a tangible set of meaning elements must be found to substitute for the entities in the outside world. The ontology in ontological semantics is the next best thing to being able to refer to the outside world directly. It is a model of that world actually constructed so that it reflects, to the best of the researcher’s ability, the outside world (including beliefs, non-existing entities, etc.). Moreover, the ontology records this knowledge not in a formal, “scientific” way but rather in a commonsense way, which, we believe, is exactly what is reflected in natural language meanings.

There are computational approaches to meaning that do not involve an overt ontological level. We

believe (and argue for it, for instance, in Nirenburg and Raskin 1996—cf. Chapter 4) that the description of meaning is more overt and complete when the metalanguage used for this task is independently and comprehensively defined.

2.6.3.3 Why should meaning be machine tractable?

This premise is rather straightforward because it is dictated by the nature of the description and applications of the theory. These descriptions and applications should be formulated so that they can be incorporated as data, heuristics or algorithms in computer programs. Machine tractability is not implied by the formality of a theory. For example, it is widely understood now, though not for a long time, that a meticulous and rigorous logical formalism of Montague grammars is not machine tractable (see Footnote 17 above) because, for one thing, it was never developed with a computer application in mind and thus lacked the necessary procedurality.

A pattern of discrepancy between theoretical and machine-tractable formalisms extends beyond semantics. Thus, attempts to develop a syntactic parser directly on the basis of early transformational syntax failed. This eventuality could be predicted if the term ‘generative’ in ‘generative grammar’ were understood in its intended mathematical—rather than procedural—sense (see Newell and Simon 1972).

2.6.3.4 Why should meaning be treated as both compositional and non-compositional?

This premise is not shared by two groups of researchers. Some philosophers of language declare their opposition to the notion of compositionality of meaning (e.g., Searle 1982b, who dismissed the phenomenon as pure ‘combinatorics’³⁵). This position also seems to follow from Wittgenstein’s anti-representationalist stance. Conversely, formal semanticists and most philosophers of language rely entirely on compositionality for producing of meaning representations. As indicated above, we hold ourselves accountable both for compositional and non-compositional aspects of text meaning, such as phrasals, deixis and pragmatic meaning, and it is the existence of both of these aspects that justifies this premise.

2.7 “Post-Empirical” Philosophy of Linguistics

In this chapter, we have argued for the need for theory as well as for the philosophy of the field underlying and determining the process of theory building. We have discussed the components of a linguistic theory and argued that distinguishing them makes the task of theory building more manageable and precise. We introduced and discussed several important parameters of theories. We then extended the discussion of the philosophical matter of theory building into applications. We finished by partially demonstrating how and why one sample parameter, albeit a crucially important one, works on a particular theory.

The experience of working on ontological semantic implementations has been critical for this effort. First, the complexity forced us to make many choices. Second, the necessity to make them

35. The paper published in the proceedings of the conference as Searle (1986) is the paper Searle had intended to deliver back in 1982. At the conference itself, however, he chose instead to deliver a philosopher’s response to Raskin (1986), the other plenary paper, attacking primarily the compositional aspect of the proposed script-based semantic theory.

in a consistent and principled way has become evident. We were in the business of creating descriptions; we were developing methodologies for producing those descriptions; and the format of the descriptions and, therefore, the nature of the methodologies, are, we had established (see Sections 2.4.2 and 2.5 above), determined by a theory. We needed to make this theory explicit, and we needed a basis for those theories and for preferring one theory over the rest at numerous junctures. All of that has led us to develop a somewhat uncommon, “post-empirical” philosophy, and we would like to comment on this briefly.

A canonical relationship between theory and practice in science is that a theory precedes an experiment (see, for instance, Hegel 1983, Kapitsa 1980). More accurately, a theoretical hypothesis is formed in the mind of the scholar and an experiment is conducted to confirm the hypothesis (or rather to fail to falsify it this time around, as Popper would have it—see 3.4 above). This kind of theory is, of course, pre-empirical, and the approach is deductive.

In reality, we know, the scientist may indeed start with the deductive theory-to-practice move but then comes back to revise the theory after the appropriate experiments in the reverse practice-to-theory move, and that move is inductive³⁶. The resulting approach is hybrid deductive-inductive, which alternates the theory-to-practice and practice-to-theory moves and leads to the theory-to-practice-to-theory-to-practice-to-theory-to-etc. string, which is interrupted when the scientist completes setting up all the general rules of the body of a theory. This is, apparently, the content of what we called post-empirical philosophy: surely, some metatheoretical premises—and, we have progressively come to believe, even broader and less strict presuppositions of a general cultural, social and historical nature—informed us before we started developing ontological semantics. But it was the process of its implementation that clarified and modified those premises and led to the specification of the theory underlying the implementation activity.

When the general rules of the body of a theory are represented as, basically, universally quantified logical propositions, such a theory falls within the 20th-century analytical tradition in philosophy. Note that ontological semantics adopts the analytical paradigm—the only one recognized by linguistics, computational linguistics, and AI—uncritically. Contrary to our principles of making explicit choices on a principled basis, we never questioned the analytical paradigm and never compared it to its major competitor in contemporary philosophy, namely, phenomenology.³⁷

The above iterative deductive-inductive sequence shows that a theory can emerge post-empirically, and commonly they do, at least, in part. What is much less common, we believe—and we made quite an effort to find a precedent for the position we propound here—is post-empirical philosophy of science. In fact, there is an ongoing conflict between the philosophers of science and scientists—or, more accurately, the active process of the two parties ignoring each other rather than engaging in explicit mutual criticism. As Moody explains, “[t]he dynamics of this collaboration are not always completely friendly. Certain philosophical conclusions may be unwelcome or

36. In contemporary psychology, unlike in science, the deductive cycle is excluded completely in favor of the inductive one. In the dominant methodology, one goes into a series of masterfully designed experiments with the so-called “null hypothesis” and strives to observe and formulate a theory from clustering the results on the basis of some form of factor analysis or a similar evidence analysis method. For the best work in the psychology of personality, for instance, see Ruch (1998) and references to his and his associates’ work there.

even unacceptable to the scientist, who may insist that she is the only one qualified to have an opinion. This is especially likely when philosophers pass harsh judgment upon some research program and its alleged findings” (1993: 5: cf. Footnote 29 above).

And this brings us to what we consider the most important result of this exercise in the philosophy of linguistics. Whether we have achieved what we set out to achieve in this chapter, there are two uncommon perspectives that we have displayed here by virtue of the post-empirical nature of our philosophy of science, as exemplified by our philosophy of linguistics. First, this philosophical proposal is offered by two practicing scientists, and the proposal emerged from practice, which effectively bridges the philosopher-scientist gap. Secondly, the practice demanded specific recommendations for significant (and tough) choices in theory building, thus pushing the philosophy of science back to the essential “big” issues it once aspired to address.

It is almost routine in contemporary philosophy itself to lament the predominance of highly sophisticated, and often outright virtuoso, discussions of intricate technical details in a single approach over consistent pursuits of major research questions. Our work seems to indicate that there are—or should be—hungry consumers in academic disciplines, whose work needs answers to these big questions, and these answers are expected to come from the philosophy of science. Should it be a centralized effort for the discipline or should every scientist do the appropriate philosophy-of-science work himself or herself as he or she goes? We have had to take the latter route, that of self-sufficiency, and we cannot help wondering if we have done it right or was it more like the Maoist attempt of the 1960s to increase the Chinese national steel production output by making every family manufacture a little steel every day after dinner in their pocket-size backyard blast furnace.

37. Unbeknownst to most scientists, including linguists, the analytical tradition has an increasingly popular competitor in phenomenology, a view of the world from the vantage point of a direct first-hand experience. University philosophy departments are usually divided into analytical and phenomenological factions, and the latter see the intensional, explanatory generalizations of the former as, basically, arbitrary leaps of faith, while the former see the anti-generalizational, extensional discussions of the latter as rather a tortuous and unreliable way to... generalizations. Uneasy peace is maintained by not talking with and not reading each other. According to Dupuy (2000: xii-xiii), this is mirrored on a much larger scale in American academia by the phenomenological (post-structuralist, postmodernist) vs. analytical split between the humanities and social sciences, respectively. Phenomenology sees itself as having been established by Hegel (1931). Analytical philosophers see it as hijacked by Heidegger (1949, 1980), and notoriously perverted by the likes of Derrida (e.g., 1967, 1987) into an easily reversible, anchorless, relativistic chaos of post-modernism that is easy for a scientist to dismiss offhand. However, the mainstream Husserlian (1964, 1982) phenomenology is a serious and respectable alternative philosophical view, except that it is hard for an analytically trained scientist to see how it can be applied. In an occasional offshoot (see, for instance, Bourdieu’s 1977 “theory of practice”), phenomenology can even be seen as coming tantalizingly close to the inductive approach within the analytical tradition. On the analytical side, Wittgenstein’s “meaning is use” and the ordinary-language philosophy (see 2.6.2.1 above) come close to the phenomenological side but can be implemented with extensional, non-representational corpus-based statistical methods.