Formalizing Natural Languages
For Nadia “Nooj” Malinovich Silberztein, the Mensch of the family, without whom neither this book, nor the project named after her, would have happened.

And for my two children, Avram and Rosa, who remind me every day of the priorities in my life.
Formalizing Natural Languages

The NooJ Approach

Max Silberztein
Contents

Acknowledgments .......................................................... xi

Chapter 1. Introduction: the Project ................................. 1
  1.1. Characterizing a set of infinite size .......................... 4
  1.2. Computers and linguistics ..................................... 5
  1.3. Levels of formalization ......................................... 6
  1.4. Not applicable .................................................. 7
    1.4.1. Poetry and plays on words .............................. 7
    1.4.2. Stylistics and rhetoric .................................. 9
    1.4.3. Anaphora, coreference resolution, and semantic disambiguation ........................................... 10
    1.4.4. Extralinguistic calculations ............................. 12
  1.5. NLP applications ............................................... 12
    1.5.1. Automatic translation .................................... 14
    1.5.2. Part-of-speech (POS) tagging ........................... 18
    1.5.3. Linguistic rather than stochastic analysis ............. 27
  1.6. Linguistic formalisms: NooJ ................................. 27
  1.7. Conclusion and structure of this book ....................... 30
  1.8. Exercises ..................................................... 31
  1.9. Internet links ................................................ 32

Part 1. Linguistic Units .................................................. 35

Chapter 2. Formalizing the Alphabet ............................... 37
  2.1. Bits and bytes ................................................ 37
  2.2. Digitizing information ....................................... 39
  2.3. Representing natural numbers ............................... 39
    2.3.1. Decimal notation ....................................... 39
Chapter 3. Defining Vocabulary ........................................ 59

3.1. Multiple vocabularies and the evolution
of vocabulary .......................................................... 59
3.2. Derivation ............................................................. 63
  3.2.1. Derivation applies to vocabulary elements ................. 63
  3.2.2. Derivations are unpredictable ................................ 64
  3.2.3. Atomicity of derived words ................................. 65
3.3. Atomic linguistic units (ALUs) ................................ 67
  3.3.1. Classification of ALUs ...................................... 67
3.4. Multiword units versus analyzable
sequences of simple words .......................................... 70
  3.4.1. Semantics ...................................................... 72
  3.4.2. Usage ......................................................... 76
  3.4.3. Transformational analysis ................................... 77
3.5. Conclusion ........................................................... 80
3.6. Exercises ............................................................. 81
3.7. Internet links ........................................................ 81

Chapter 4. Electronic Dictionaries ......................................... 83

4.1. Could editorial dictionaries be reused? ........................... 83
4.2. LADL electronic dictionaries ....................................... 90
  4.2.1. Lexicon-grammar ............................................ 90
  4.2.2. DELA ....................................................... 93
4.3. Dubois and Dubois-Charlier electronic dictionaries .......... 94
  4.3.1. The Dictionnaire électronique des mots .................... 95
  4.3.2. Les Verbes Français (LVF) ................................ 97
4.4. Specifications for the construction of
an electronic dictionary ............................................... 99
4.4.1. One ALU = one lexical entry .................................. 99
4.4.2. Importance of derivation ......................................... 100
4.4.3. Orthographic variation ............................................ 101
4.4.4. Inflection of simple words, compound words, and expressions ........................................................................ 103
4.4.5. Expressions .............................................................. 104
4.4.6. Integration of syntax and semantics .................................. 104
4.5. Conclusion ................................................................. 107
4.6. Exercises .................................................................. 108
4.7. Internet links ............................................................... 108

Part 2. Languages, Grammars and Machines ................................ 111

Chapter 5. Languages, Grammars, and Machines ...................... 113
5.1. Definitions ................................................................. 113
  5.1.1. Letters and alphabets .................................................. 113
  5.1.2. Words and languages ................................................ 114
  5.1.3. ALU, vocabularies, phrases, and languages ......................... 114
  5.1.4. Empty string ........................................................... 115
  5.1.5. Free language .......................................................... 116
  5.1.6. Grammars ............................................................... 116
  5.1.7. Machines ............................................................... 117
5.2. Generative grammars ...................................................... 118
5.3. Chomsky-Schützenberger hierarchy ................................... 119
  5.3.1. Linguistic formalisms ................................................ 122
5.4. The NooJ approach ......................................................... 124
  5.4.1. A multifaceted approach ............................................. 124
  5.4.2. Unified notation ...................................................... 125
  5.4.3. Cascading architecture .............................................. 127
5.5. Conclusion .................................................................. 127
5.6. Exercises .................................................................. 128
5.7. Internet links ............................................................... 129

Chapter 6. Regular Grammars ............................................... 131
6.1. Regular expressions ...................................................... 131
  6.1.1. Some examples of regular expressions ......................... 135
6.2. Finite-state graphs ....................................................... 137
6.3. Non-deterministic and deterministic graphs ....................... 139
6.4. Minimal deterministic graphs ....................................... 141
6.5. Kleene’s theorem .......................................................... 142
6.6. Regular expressions with outputs and finite-state transducers ................................................................. 146
6.7. Extensions of regular grammars ........................................... 151
  6.7.1. Lexical symbols ....................................................... 151
  6.7.2. Syntactic symbols ................................................... 153
  6.7.3. Symbols defined by grammars ...................................... 154
  6.7.4. Special operators .................................................... 155
  6.8. Conclusion .............................................................. 159
  6.9. Exercises ............................................................... 159
  6.10. Internet links ......................................................... 159

Chapter 7. Context-Free Grammars ........................................... 161

  7.1. Recursion ............................................................... 164
    7.1.1. Right recursion .................................................. 166
    7.1.2. Left recursion ................................................... 167
    7.1.3. Middle recursion ................................................ 168
  7.2. Parse trees ........................................................... 170
  7.3. Conclusion ............................................................. 173
  7.4. Exercises ............................................................. 173
  7.5. Internet links ......................................................... 174

Chapter 8. Context-Sensitive Grammars ................................... 175

  8.1. The NooJ approach ..................................................... 176
    8.1.1. The anbncn language ............................................. 177
    8.1.2. The language a^n ................................................ 180
    8.1.3. Handling reduplications ........................................ 181
    8.1.4. Grammatical agreements ........................................ 182
    8.1.5. Lexical constraints in morphological grammars ............. 185
  8.2. NooJ contextual constraints ........................................ 186
  8.3. NooJ variables ........................................................ 188
    8.3.1. Variables’ scope ................................................ 188
    8.3.2. Computing a variable’s value .................................. 189
    8.3.3. Inheriting a variable’s value .................................. 191
  8.4. Conclusion ............................................................. 191
  8.5. Exercises ............................................................. 192
  8.6. Internet links ........................................................ 192

Chapter 9. Unrestricted Grammars ......................................... 195

  9.1. Linguistic adequacy .................................................. 197
  9.2. Conclusion ............................................................. 199
  9.3. Exercise ............................................................... 199
  9.4. Internet links ........................................................ 199
Part 3. Automatic Linguistic Parsing ................................. 201

Chapter 10. Text Annotation Structure .............................. 205

  10.1. Parsing a text ........................................... 205
  10.2. Annotations .............................................. 206
    10.2.1. Limits of XML/TEI representation ..................... 207
  10.3. Text annotation structure (TAS) .......................... 208
  10.4. Exercise ............................................... 211
  10.5. Internet links .......................................... 212

Chapter 11. Lexical Analysis .......................................... 213

  11.1. Tokenization ............................................. 213
    11.1.1. Letter recognition .................................... 214
    11.1.2. Apostrophe/quote ..................................... 217
    11.1.3. Dash/hyphen ....................................... 219
    11.1.4. Dot/period/point ambiguity ........................... 222
  11.2. Word forms .............................................. 224
    11.2.1. Space and punctuation ............................... 224
    11.2.2. Numbers ........................................... 226
    11.2.3. Words in upper case ................................ 228
  11.3. Morphological analyses .................................. 229
    11.3.1. Inflectional morphology ............................. 230
    11.3.2. Derivational morphology ............................ 234
    11.3.3. Lexical morphology ................................ 236
    11.3.4. Agglutinations ..................................... 239
  11.4. Multiword unit recognition ................................ 241
  11.5. Recognizing expressions .................................. 243
    11.5.1. Characteristic constituent ........................... 244
    11.5.2. Varying the characteristic constituent ............... 245
    11.5.3. Varying the light verb ................................ 246
    11.5.4. Resolving ambiguity ................................ 247
    11.5.5. Annotating expressions ................................ 251
  11.6. Conclusion .............................................. 254
  11.7. Exercise ............................................... 255

Chapter 12. Syntactic Analysis ....................................... 257

  12.1. Local grammars .......................................... 257
    12.1.1. Named entities ...................................... 257
    12.1.2. Grammatical word sequences .......................... 262
    12.1.3. Automatically identifying ambiguity ................... 263
  12.2. Structural grammars ..................................... 265
12.2.1. Complex atomic linguistic units ............................... 266
12.2.2. Structured annotations ........................................... 268
12.2.3. Ambiguities ....................................................... 270
12.2.4. Syntax trees vs parse trees ...................................... 273
12.2.5. Dependency grammar and tree .................................. 276
12.2.6. Resolving ambiguity transparently .............................. 279
12.3. Conclusion ............................................................ 280
12.4. Exercises ............................................................... 281
12.5. Internet links ............................................................ 281

Chapter 13. Transformational Analysis ................................. 283
13.1. Implementing transformations ....................................... 286
13.2. Theoretical problems .................................................. 292
  13.2.1. Equivalence of transformation sequences ...................... 292
  13.2.2. Ambiguities in transformed sentences ......................... 293
  13.2.3. Theoretical sentences ............................................. 294
  13.2.4. The number of transformations to be implemented ............ 295
13.3. Transformational analysis with NooJ ................................. 297
  13.3.1. Applying a grammar in “generation” mode .................... 298
  13.3.2. The transformation’s arguments .................................. 299
13.4. Question answering .................................................... 303
13.5. Semantic analysis ..................................................... 304
13.6. Machine translation ................................................... 305
13.7. Conclusion ............................................................... 309
13.8. Exercises ............................................................... 309
13.9. Internet links ............................................................ 310

Conclusion ................................................................. 311

Bibliography ............................................................... 315

Index ................................................................. 327
Acknowledgments

I would like to thank the University of Franche-Comté and my colleagues in the ELLIADD laboratory for believing in the NooJ project and supporting the community of NooJ users unfailingly since its inception.

It would be impossible for me to mention every single one of the colleagues and students who have participated, in one way or another, in the extremely ambitious project described in this book – that of formalizing natural languages! The NooJ software has been in use since 2002 by a community of researchers and students; see www.nooj4nlp.net. NooJ was developed in direct cooperation with all its users who devoted their energy to this or that specific problem, or to one language or another. Spelling in Semitic languages, variation in Asian languages, intonation in Armenian, inflection in Hungarian, phrasal verbs in English, derivation in Slavic languages, composition in Greek and in Germanic languages, etc. pose a wide variety of linguistic problems, and without the high standards of these linguists the NooJ project would never have known the success it is experiencing today. Very often, linguistic questions that seemed “trivial” at the time have had a profound influence on the development of NooJ.

Among its users, there are some “NooJ experts” to whom I would like to give particular thanks, as they participated directly in its design, and had the patience to help me with long debugging sessions. I thank them for their ambition and their patience: Héla Fehri, Kristina Kocijan, Slim Mesfär, Cristina Mota, and Simonetta Vietri.
I would also like to thank Danielle Leeman and François Trouilleux for their detailed review of the original book, and Peter Machonis for his review of the English version, as well as for verifying the relevance of the English examples, which contributed greatly to the quality of this book.

Max SILBERZTEIN
November, 2015.
Introduction: the Project

The project described in this book is at the very heart of linguistics; its goal is to describe, exhaustively and with absolute precision, all the sentences of a language likely to appear in written texts. This project fulfills two needs: it provides linguists with tools to help them describe languages exhaustively (linguistics), and it aids in the building of software able to automatically process texts written in natural language (natural language processing, or NLP).

A linguistic project needs to have a theoretical and methodological framework (how to describe this or that linguistic phenomenon; how to organize the different levels of description); formal tools (how to write each description); development tools to test and manage each description; and engineering tools to be used in sharing, accumulating, and maintaining large quantities of linguistic resources.

There are many potential applications of descriptive linguistics for NLP: spell-checkers, intelligent search engines, information extractors and annotators, automatic summary producers, automatic translators, etc. These applications have the potential for considerable economic usefulness, and it is therefore important for linguists to make use of these technologies and to be able to contribute to them.

1 Non-written languages, such as speech or sign language, can be transcribed by using specific alphabets, such as the International Phonetic Alphabet or the American Sign Alphabet. The resulting transcribed text indeed constitutes a written text.

2 [SAU 16, BLO 33] were among the first to attempt to rationalize the description of languages.
For now, we must reduce the overall linguistic project of describing all phenomena related to the use of language, to a much more modest project: here, we will confine ourselves to seeking to describe the set of all of the sentences that may be written or read in natural-language texts. The goal, then, is simply to design a system capable of distinguishing between the two sequences below:

a) *Joe is eating an apple*

b) *Joe eating apple is an*

Sequence (a) is a grammatical sentence, while sequence (b) is not.

This project constitutes the mandatory foundation for any more ambitious linguistic projects. Indeed it would be fruitless to attempt to formalize text styles (stylistics), the evolution of a language across the centuries (etymology), variations in a language according to social class (sociolinguistics), cognitive phenomena involved in the learning or understanding of a language (psycholinguistics), etc. without a model, even a rudimentary one, capable of characterizing sentences.

If the number of sentences were finite – that is, if there were a maximum number of sentences in a language – we would be able to list them all and arrange them in a database. To check whether an arbitrary sequence of words is a sentence, all we would have to do is consult this database: it is a sentence if it is in the database, and otherwise it is not. Unfortunately, there are an infinite number of sentences in a natural language. To convince ourselves of this, let us resort to a *redictio ad absurdum*: imagine for a moment that there are $n$ sentences in English.

Based on this finite number $n$ of initial sentences, we can construct a second set of sentences by putting the sequence *Lea thinks that*, for example, before each of the initial sentences:

\[
\text{Joe is sleeping} \rightarrow \text{Lea thinks that Joe is sleeping}
\]

\[
\text{The party is over} \rightarrow \text{Lea thinks that the party is over}
\]

Using this simple mechanism, we have just doubled the number of sentences, as shown in the figure below.
This mechanism can be generalized by using verbs other than the verb *to think*; for example:

*Lea (believes | claims | dreams | knows | realizes | thinks | ...) that Sentence.*

There are several hundred verbs that could be used here. Likewise, we could replace *Lea* with several thousand human nouns:

*(The CEO | The employee | The neighbor | The teacher | ...) thinks that Sentence.*

Whatever the size *n* of an initial set of sentences, we can thus construct \( n \times 100 \times 1,000 \) sentences simply by inserting before each of the initial sentences, sequences such as *Lea thinks that*, *Their teacher claimed that*, *My neighbor declared that*, etc.

Language has other mechanisms that can be used to expand a set of sentences exponentially. For example, based on *n* initial sentences, we can construct \( n \times n \) sentences by combining all of these sentences in pairs and inserting the word *and* between them. For example:

*It is raining + Joe is sleeping → It is raining and Joe is sleeping*
This mechanism can also be generalized by using several hundred connectors; for example:

*It is raining (but | nevertheless | therefore | where | while | ...) Joe is sleeping.*

These two mechanisms (linking of sentences and use of connectors) can be used multiple times in a row, as in the following:

*Lea claims that Joe hoped that Ida was sleeping.*

*It was raining while Lea was sleeping, however Ida is now waiting, but the weather should clear up as soon as night falls.*

Thus these mechanisms are said to be *recursive*; the number of sentences that can be constructed with recursive mechanisms is infinite. Therefore it would be impossible to define all of these sentences *in extenso*. Another way must be found to characterize the set of sentences.

### 1.1. Characterizing a set of infinite size

Mathematicians have known for a long time how to define sets of infinite size. For example, the two rules below can be used to define the set of all natural numbers $\mathbb{N}$:

- (a) Each of the ten elements of set \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} is a natural number;
- (b) any word that can be written as $xy$ is a natural number if and only if its two constituents $x$ and $y$ are natural numbers.

These two rules constitute a formal definition of all natural numbers. They make it possible to distinguish natural numbers from any other object (decimal numbers or others). For example:

- Is the word “123” a natural number? Thanks to rule (a), we know that “1” and “2” are natural numbers. Rule (b) allows us to deduce from this that “12” is a natural number. Thanks to rule (a) we know that “3” is a natural number; since “12” and “3” are natural numbers, then rule (b) allows us to deduce that “123” is a natural number.
– The word “2.5” is not a natural number. Rule (a) enables us to deduce that “2” is a natural number, but it does not apply to the decimal point “.”. Rule (b) can only apply to two natural numbers, therefore it does not apply to the decimal point because it is not a natural number. In this case, “2.” is not a natural number; therefore “2.5” is not a natural number either.

There is an interesting similarity between this definition of set $\mathbb{N}$ and the problem of characterizing the sentences in a language:

– Rule (a) describes *in extenso* the finite set of numerals that must be used to form valid natural numbers. This rule resembles a dictionary in which we would list all the words that make up the vocabulary of a language.

– Rule (b) explains how numerals can be combined to construct an infinite number of natural numbers. This rule is similar to grammatical rules that specify how to combine words in order to construct an infinite number of sentences.

To describe a natural language, then, we will proceed as follows: firstly we will define *in extenso* the finite number of basic units in a language (its vocabulary); and secondly, we will list the rules used to combine the vocabulary elements in order to construct sentences (its grammar).

### 1.2. Computers and linguistics

Computers are a vital tool for this linguistic project, for at least four reasons:

– From a theoretical point of view, a computer is a device that can verify automatically that an element is part of a mathematically-defined set. Our goal is then to construct a device that can automatically verify whether a sequence of words is a valid sentence in a language.

– From a methodological point of view, the computer will impose a framework to describe linguistic objects (words, for example) as well as the rules for use of these objects (such as syntactic rules). The way in which linguistic phenomena are described must be consistent with the system: any inconsistency in a description will inevitably produce an error (or “bug”).
When linguistic descriptions have been entered into a computer, a computer can apply them to very large texts in order to extract from these texts examples or counterexamples that validate (or not) these descriptions. Thus a computer can be used as a scientific instrument (this is the corpus linguistics approach), as the telescope is in astronomy or the microscope in biology.

Describing a language requires a great deal of descriptive work; software is used to help with the development of databases containing numerous linguistic objects as well as numerous grammar rules, much like engineers use computer-aided design (CAD) software to design cars, electronic circuits, etc. from libraries of components.

Finally, the description of certain linguistic phenomena makes it possible to construct NLP software applications. For example, if we have a complete list of the words in a language, we can build a spell-checker; if we have a list of rules of conjugation we can build an automatic conjugator. A list of morphological and phonological rules also makes it possible to suggest spelling corrections when the computer has detected errors, while a list of simple and compound terms can be used to build an automatic indexier. If we have bilingual dictionaries and grammars we can build an automatic translator, and so forth. Thus the computer has become an essential tool in linguistics, so much so that opposing “computational linguists” with “pure linguists” no longer makes sense.

1.3. Levels of formalization

When we characterize a phenomenon using mathematical rules, we formalize it. The formalization of a linguistic phenomenon consists of describing it, by storing both linguistic objects and rules in a computer. Languages are complicated to describe, partly because interactions between their phonological and writing systems have multiplied the number of objects to process, as well as the number of levels of combination rules. We can distinguish five fundamental levels of linguistic phenomena; each of these levels corresponds to a level of formalization.

---

To analyze a written text, we access letters of the alphabet rather than words; thus it is necessary to describe the link between the alphabet and the orthographic forms we wish to process (spelling). Next, we must establish a link between the
orthographic forms and the corresponding vocabulary elements (morphology). Vocabulary elements are generally listed and described in a lexicon that must also show all potential ambiguities (lexicography). Vocabulary elements combine to build larger units such as phrases which then combine to form sentences; therefore rules of combination must be established (syntax). Finally, links between elements of meaning which form a predicate transcribed into an elementary sentence, as well as links between predicates in a complex sentence, must be established (semantics).

1.4. Not applicable

We do not always use language to represent and communicate information directly and simply; sometimes we play with language to create sonorous effects (for example in poetry). Sometimes we play with words, or leave some “obvious” information implicit because it stems from the culture shared by the speakers (anaphora). Sometimes we express one idea in order to suggest another (metaphor). Sometimes we use language to communicate statements about the real world or in scientific spheres, and sometimes we even say the opposite of what we really mean (irony).

It is important to clearly distinguish problems that can be solved within a strictly linguistic analytical framework from those that require access to information from other spheres in order to be solved.

1.4.1. Poetry and plays on words

Writers, poets, and authors of word games often take the liberty of constructing texts that violate the syntactic or semantic constraints of language. For example, consider the following text:

For her this rhyme is penned, whose luminous eyes
Brightly expressive as the twins of Leda,
Shall find her own sweet name, that nesting lies,
Upon the page, enwrapped from every reader.

3 From the poem A Valentine by Edgar Allan Poe.
This poem is an acrostic, meaning that it contains a puzzle which readers are invited to solve. We cannot rely on linguistic analysis to solve this puzzle. But, to even understand that the poem is a puzzle, the reader must figure out that this rhyme refers to the poem itself. Linguistic analysis is not intended to figure out what in the world this rhyme might be referring to; much less to decide among the possible candidates.

... luminous eyes brightly expressive as the twins of Leda ...

The association between the adjective luminous and eyes is not a standard semantic relationship; unless the eyes belong to a robot, eyes are not luminous. This association is, of course, metaphorical: we have to understand that luminous eyes means that the owner of the eyes has a luminous intelligence, and that we are perceiving this luminous intelligence by looking at her eyes.

The twins of Leda are probably the mythological heroes Castor and Pollux (the twin sons of Leda, the wife of the king of Sparta), but they are not particularly known for being expressive. These two heroes gave their names to the constellation Gemini, but I confess that I do not understand what an expressive constellation might be. I suspect the author rather meant to write:

... expressive eyes brightly luminous as the twins of Leda ...

The associations between the noun name and the verbal forms lies, nestling, and enwrapped are no more direct; we need to understand that it is the written form of the name which is present on the physical page where the poem is written, and that it is hidden from the reader.

If we wish to make a poetic analysis of this text, the first thing to do is thus to note these non-standard associations, so we will know where to run each poetic interpretive analysis. But if we do not even know that eyes are not supposed to be luminous, we will not be able to even figure out that there is a metaphor, therefore we will not be able to solve it (i.e. to compute that the woman in question is intelligent), and so we will have missed an important piece of information in the poem. More generally, in order to understand a poem’s meaning, we must first note the semantic violations it contains. To do this, we need a linguistic model capable of distinguishing
“standard” associations such as an intelligent woman, a bright constellation, a name written on a page, etc. from associations requiring poetic analysis, such as luminous eyes, an expressive constellation, a name lying upon a page.

Analyzing poems can pose other difficulties, particularly at the lexical and syntactic levels. In standard English, word order is less flexible than in poems. To understand the meaning of this poem, a modern reader has to start by rewriting (in his or her mind) the text in standard English, for example as follows:

This rhyme is written for her, whose luminous eyes (as brightly expressive as the twins of Leda) will find her own sweet name, which lies on the page, nestling, enwrapped from every reader.

The objective of the project described in this book is to formalize standard language without solving poetic puzzles, or figuring out possible referents, or analyzing semantically non-standard associations.

1.4.2. Stylistics and rhetoric

Stylistics studies ways of formulating sentences in speech. For example, in a text we study the use of understatements, metaphors, and metonymy (“figures of style”), the order of the components of a sentence and that of the sentences in a speech, and the use of anaphora. Here are a few examples of stylistic phenomena that cannot be processed in a strictly linguistic context:

Understatement: Joe was not the fastest runner in the race

Metaphor: The CEO is a real elephant

Metonymy: The entire table burst into laughter

In reality, the sentence Joe was not the fastest runner in the race could mean here that Joe came in last; so, in a way, this sentence is not saying what it is expressing! Unless we know the result of the race, or have access to information about the real Joe, we cannot expect a purely linguistic analysis system to detect understatements, irony or lies.
To understand the meaning of the sentence *The CEO is a real elephant*, we need to know firstly that a CEO cannot really be an elephant, and therefore that this is a metaphor. Next we need to figure out which “characteristic property” of elephants is being used in the metaphor. Elephants are known for several things: they are big, strong, and clumsy; they have long memories; they are afraid of mice; they are an endangered species; they have big ears; they love to take mud-baths; they live in Africa or India, etc. Is the CEO clumsy? Is he/she afraid of mice? Does he/she love mud-baths? Does he/she have a good memory? To understand this statement, we would have to know the context in which the sentence was said, and we might also need to know more about the CEO in question.

To understand the meaning of the sentence *The entire table burst into laughter*, it is necessary first to know that a table is not really capable of bursting into laughter, and then to infer that there are people gathered around a table (during a meal or a work meeting) and that it is these people who burst out laughing. The noun *table* is neither a collective human noun (such as *group* or *colony*), nor a place that typically contains humans (such as *meeting room* or *restaurant*), nor an organization (such as *association* or *bank*); therefore using only the basic lexical properties associated with the noun *table* will not be enough to comprehend the sentence.

It is quite reasonable to expect a linguistic system to detect that the sentences *The CEO is a real elephant* and *The entire table burst into laughter* are not standard sentences; for example, by describing *CEO* as a human noun, describing *table* as a concrete noun, and requiring *to burst into laughter* to have a human subject, we can learn from a linguistic analysis that these sentences are not “standard”, and that it is therefore necessary to initiate an extra-linguistic computation such as metaphor or metonymy calculations in order to interpret them.

The linguistic project described in this book is not intended to solve understatements, metaphors, or metonymy, but it must be able to detect sentences that are deviant in comparison to the standard language.

### 1.4.3. Anaphora, coreference resolution, and semantic disambiguation

Coreference: *Lea invited Ida for dinner. She brought a bottle of wine.*
Anaphora: Phelps returned. The champion brought back 6 medals with him.

Semantic ambiguity: The round table is in room B17.

In order to understand that in the sentence She brought a bottle of wine, she refers to Ida and not Lea, we need to know that it is usually the guest who travels and brings a bottle of wine. This social convention is commonplace throughout the modern Western world, but we would need to be sure that this story does not take place in a society where it is the person who invites who brings beverages.

In order to understand that The champion is a reference to Phelps, we have to know that Phelps is a champion. Note that dozens of other nouns could have been used in this anaphora: the American, the medal-winner, the record-holder, the swimming superstar, the young man, the swimmer, the former University of Florida student, the breakaway, the philanthropist, etc.

In order to eliminate the ambiguity of the sequence round table (between “a table with a round shape” and “a meeting”), we would need to have access to a wider context than the sentence alone.

The linguistic project described in this book is not intended to resolve anaphora or semantic ambiguities.

NOTE. – I am not saying that it is impossible to process poetry, word games, understatements, metaphors, metonymy, coreference, anaphora, and semantic ambiguities; I am only saying that these phenomena lie outside the narrow context of the project presented in this book. There are certainly “lucky” cases in which linguistic software can automatically solve some of these phenomena. For example, in the following sequence:

Joe invited Lea for dinner. She brought a bottle of wine

a simple verification of the pronoun’s gender would enable us to connect She to Lea. Conversely, it is easy to build software which, based on the two sentences Joe invited Lea to dinner and Lea brought a bottle of wine, would produce the sentence She brought a bottle of wine. Likewise, in the sentence:

The round table is taking place in room B17
a linguistic parser could automatically figure out that the noun *round table* refers to a meeting, provided that it has access to a dictionary in which the noun *round table* is described as being an abstract noun (synonymous with *meeting*), and the verb *to take place* is described as calling for an abstract subject.

### 1.4.4. Extralinguistic calculations

Consider the following statements:

a) *Two dollars plus three dollars make four dollars.*

b) *Clinton was already president in 1536.*

c) *The word God has four letters.*

d) *This sentence is false.*

These statements are expressed using sentences that are well-formed because they comply with the spelling, morphological, syntactic, and semantic rules of the English language. However, they express statements that are incorrect in terms of mathematics (a), history (b), spelling (c), or logic (d). To detect these errors we would need to access knowledge that is not part of our strictly linguistic project\(^4\).

The project described in this book is confined to the formalization of language, without taking into account speakers’ knowledge about the real world.

### 1.5. NLP applications

Of course, there are fantastic software applications capable of processing extralinguistic problems! For example, the IBM computer Watson won on the game show *Jeopardy!* in spectacular fashion in 2011; I have a lot of fun

\(^4\) Some linguists have put forward meta-linguistics examples of type (c) to demonstrate the necessity of using unrestricted grammar to describe languages. In addition, many NLP researchers use phenomena such as metaphors, resolution of anaphora, detection of understatement, etc. to demonstrate the inadequacy of linguistic techniques to handle… what I consider extralinguistic phenomena.
asking my smart watch questions. In the car, I regularly ask Google Maps to guide me verbally to my destination; my language-professor colleagues have trouble keeping their students from using Google Translate; and the subtitles added automatically to YouTube videos are a precious resource for people who are hard of hearing [GRE 11], etc.

All of these software platforms have a NLP part, which analyzes or produces a written or verbal statement, often accompanied by a specialized module, for example a search engine or GPS navigation software. It is important to distinguish between these components: just because we are impressed by the fact that Google Maps gives us reliable directions, it does not mean it speaks perfect English. It is very possible that IBM Watson can answer a question correctly without having really “understood” the question. Likewise, a software platform might automatically summarize a text using simple techniques to filter out words, phrases or sentences it judges to be unimportant [MAN 01]5. Word-recognition systems use signal processing techniques to produce a sequence of phonemes and then determine the most “probable” corresponding sequence of words by comparing it to a reference database [JUR 00]6, etc.

Most pieces of NLP software actually produce spectacular, almost magical results, with a very low degree of linguistic competence. To produce these results, the software uses “tricks”, often based on statistical methods [MAN 99].

Unfortunately, the success of these software platforms is often used in order to show that statistical techniques have made linguistics unnecessary7. It is important, then, to understand their limitations. In the next two sections I will take a closer look at the performances of the two “flagship” statistical NLP software platforms: automatic translation and part-of-speech tagging.

5 [MAN 01] introduces the field of automatic summarizing and its issues, in particular how to detect the “important” information that should be included in a summary.
6 For the exception that proves the rule, see [RAY 06].
7 For an amusing read about the ravages of statistics in fields other than linguistics, see [SMI 14].
It understands what you say. It knows what you mean.

Talk to Siri as you would to a person. Say something like “Tell my wife I’m running late” or “Remind me to call the vet.” Siri not only understands what you say, it’s smart enough to know what you mean. So when you ask “Any good burger joints around here?” Siri will reply “I found a number of burger restaurants near you.” Then you can say “Hmm. How about tacos?” Siri remembers that you just asked about restaurants, so it will look for Mexican restaurants in the neighborhood. And Siri is proactive, so it will question you until it finds what you’re looking for.

Figure 1.2. Really?

1.5.1. Automatic translation

Today, the best-known translation software platforms use statistical techniques to suggest a translation of texts. These software platforms are regularly cited as examples of the success of statistical techniques, and everyone has seen a “magical” translation demo. It is not surprising, therefore, that most people think the problem of translation has already been solved. I do not agree.

8 For example, here is a demonstration of the instantaneous translation used by Google Glasses: www.youtube.com/watch?v=pZKWW3rzT2Q.
For example, Figure 1.3 shows how Google Translate has translated part of an article in Vietnamese (www.thanhnieunnews.com, October 2014). The text produced does not make very much sense; what does “I’m talking to him on the Navy’s generals” mean? The translated text even contains incorrect constructions (“could very well shielded”, for example).

Figure 1.4 allows us to compare Google Translate’s results with those obtained using a simple Arabic–English dictionary and a very basic translation grammar; see [BAN 15]. For example, the first sentence was wrongly translated by Google Translate as The man who went down to work instead of The man who fell went to work. Google Translate was also wrong about the second sentence, which means The man that you knew went to work and not I knew a man who went to work.
When translating languages that are more similar, such as French into English, the results produced by Google Translate are helpful, but still could not be used in a professional context or to translate a novel, a technical report, or even a simple letter, and especially not when submitting a résumé.