1

An Overview of Language Processing

1.1 Linguistics and Language Processing

Linguistics is the study and the description of human languages. Linguistic theories on grammar and meaning have been developed since ancient times and the Middle Ages. However, modern linguistics originated at the end of the nineteenth century and the beginning of the twentieth century. Its founder and most prominent figure was probably Ferdinand de Saussure (1916). Over time, modern linguistics has produced an impressive set of descriptions and theories.

Computational linguistics is a subset of both linguistics and computer science. Its goal is to design mathematical models of language structures enabling the automation of language processing by a computer. From a linguist's viewpoint, we can consider computational linguistics as the formalization of linguistic theories and models or their implementation in a machine. We can also view it as a means to develop new linguistic theories with the aid of a computer.

From an applied and industrial viewpoint, language and speech processing, which is sometimes referred to as natural language processing (NLP) or natural language understanding (NLU), is the mechanization of human language faculties. People use language every day in conversations by listening and talking, or by reading and writing. It is probably our preferred mode of communication and interaction. Ideally, automated language processing would enable a computer to understand texts or speech and to interact accordingly with human beings.

Understanding or translating texts automatically and talking to an artificial conversational assistant are major challenges for the computer industry. Although this final goal has not been reached yet, in spite of constant research, it is being approached every day, step-by-step. Even if we have missed Stanley Kubrick's prediction of talking electronic creatures in the year 2001, language processing and understanding techniques have already achieved results ranging from very promising to near perfect. The description of these techniques is the subject of this book.

1.2 Applications of Language Processing

At first, language processing is probably easier understood by the description of a result to be attained rather than by the analytical definition of techniques. Ideally, language processing would enable a computer to analyze huge amounts of text and to understand them; to communicate with us in a written or a spoken way; to capture our words whatever the entry mode: through a keyboard or through a speech recognition device; to parse our sentences; to understand our utterances, to answer our questions, and possibly to have a discussion with us – the human beings.

Language processing has a history nearly as old as that of computers and comprises a large body of work. However, many early attempts remained in the stage of laboratory demonstrations or simply failed. Significant applications have been slow to come, and they are still relatively scarce compared with the universal deployment of some other technologies such as operating systems, databases, and networks. Nevertheless, the number of commercial applications or significant laboratory prototypes embedding language processing techniques is increasing. Examples include:

- Spelling and grammar checkers. These programs are now ubiquitous in text processors, and hundred of millions of people use them every day. Spelling checkers are based on computerized dictionaries and remove most misspellings that occur in documents. Grammar checkers, although not perfect, have improved to a point that many users could not write a single e-mail without them. Grammar checkers use rules to detect common grammar and style errors (Jensen et al. 1993).
- Text indexing and information retrieval from the Internet. These programs are among the most popular of the Web. They are based on spiders that visit Internet sites and that download texts they contain. Spiders track the links occurring on the pages and thus explore the Web. Many of these systems carry out a full text indexing of the pages. Users ask questions and text retrieval systems return the Internet addresses of documents containing words of the question. Using statistics on words or popularity measures, text retrieval systems are able to rank the documents (Salton 1988, Brin and Page 1998).
- Speech dictation of letters or reports. These systems are based on speech recognition. Instead of typing using a keyboard, speech dictation systems allow a user to dictate reports and transcribe them automatically into a written text. Systems like IBM's ViaVoice have a high performance and recognize English, French, German, Spanish, Italian, Japanese, Chinese, etc. Some systems transcribe radio and TV broadcast news with a word-error rate lower than 10% (Nguyen et al. 2004).
- Voice control of domestic devices such as videocassette recorders or disc changers (Ball et al. 1997). These systems aim at being embedded in objects to provide them with a friendlier interface. Many people find electronic devices complicated and are unable to use them satisfactorily. How many of us are tape recorder illiterates? A spoken interface would certainly be an easier means to control them. Although there are many prototypes, few systems are commercially available yet. One challenge they still have to overcome is to operate in noisy environments that impair speech recognition.

- Interactive voice response applications. These systems deliver information over the telephone using speech synthesis or prerecorded messages. In more traditional systems, users interact with the application using touch-tone telephones. More advanced servers have a speech recognition module that enables them to understand spoken questions or commands from users. Early examples of speech servers include travel information and reservation services (Mast et al. 1994, Sorin et al. 1995). Although most servers are just interfaces to existing databases and have limited reasoning capabilities, they have spurred significant research on dialogue, speech recognition and synthesis.
- Machine translation. Research on machine translation is one of the oldest domains of language processing. One of its outcomes is the venerable SYSTRAN program that started with translations between English and Russian. Since then, SYSTRAN has been extended to many other languages. Another pioneer example is the *Spoken Language Translator* that translated spoken English into spoken Swedish in a restricted domain in real time (Agnäs et al. 1994, Rayner et al. 2000).
- Conversational agents. Conversational agents are elaborate dialogue systems that have understanding faculties. An example is TRAINS that helps a user plan a route and the assembling trains: boxcars and engines to ship oranges from a warehouse to an orange juice factory (Allen et al. 1995). Ulysse is another example that uses speech to navigate into virtual worlds (Godéreaux et al. 1996, Godéreaux et al. 1998).

Some of these applications are widespread, like spelling and grammar checkers. Others are not yet ready for an industrial exploitation or are still too expensive for popular use. They generally have a much lower distribution. Unlike other computer programs, results of language processing techniques rarely hit a 100% success rate. Speech recognition systems are a typical example. Their accuracy is assessed in statistical terms. Language processing techniques become mature and usable when they operate above a certain precision and at an acceptable cost. However, common to these techniques is that they are continuously improving and they are rapidly changing our way of interacting with machines.

1.3 The Different Domains of Language Processing

Historically linguistics has been divided into disciplines or levels, which go from sounds to meaning. Computational processing of each level involves different techniques such as signal and speech processing, statistics, pattern recognition, parsing, first-order logic, and automated reasoning.

A first discipline of linguistics is **phonetics**. It concerns the production and perception of acoustic sounds that form the speech signal. In each language, sounds can be classified into a finite set of **phonemes**. Traditionally, they include **vowels**: *a*, *e*, *i*, *o*; and **consonants**: *p*, *f*, *r*, *m*. Phonemes are assembled into **syllables**: *pa*, *pi*, *po*, to build up the words.

A second level concerns the **words**. The word set of a language is called a **lexicon**. Words can appear under several forms, for instance, the singular and the plural forms. **Morphology** is the study of the structure and the forms of a word. Usually a lexicon consists of root words. Morphological rules can modify or transform the root words to produce the whole vocabulary.

Syntax is a third discipline in which the order of words in a sentence and their relationships is studied. Syntax defines word categories and functions. Subject, verb, object is a sequence of functions that corresponds to a common order in many European languages including English and French. However, this order may vary, and the verb is often located at the end of the sentence in German. **Parsing** determines the structure of a sentence and assigns functions to words or groups of words.

Semantics is a fourth domain of linguistics. It considers the meaning of words and sentences. The concept of "meaning" or "signification" can be controversial. Semantics is differently understood by researchers and is sometimes difficult to describe and process. In a general context, semantics could be envisioned as a medium of our thought. In applications, semantics often corresponds to the determination of the sense of a word or the representation of a sentence in a logical format.

Pragmatics is a fifth discipline. While semantics is related to universal definitions and understandings, pragmatics restricts it – or complements it – by adding a contextual interpretation. Pragmatics is the meaning of words and sentences in specific situations.

The production of language consists of a stream of sentences that are linked together to form a **discourse**. This discourse is usually aimed at other people who can answer – it is to be hoped – through a **dialogue**. A dialogue is a set of linguistic interactions that enables the exchange of information and sometimes eliminates misunderstandings or ambiguities.

1.4 Phonetics

Sounds are produced through vibrations of the vocal cords. Several cavities and organs modify vibrations: the vocal tract, the nose, the mouth, the tongue, and the teeth. Sounds can be captured using a microphone. They result in signals such as that in Fig. 1.1.

Fig. 1.1. A speech signal corresponding to *This is* [ðɪs ɪz].

A speech signal can be sampled and digitized by an analog-to-digital converter. It can then be processed and transformed by a Fourier analysis (FFT) in a moving window, resulting in spectrograms (Figs. 1.2 and 1.3). Spectrograms represent the distribution of speech power within a frequency domain ranging from 0 to 10,000 Hz over time. This frequency domain corresponds roughly to the sound production possibilities of human beings.

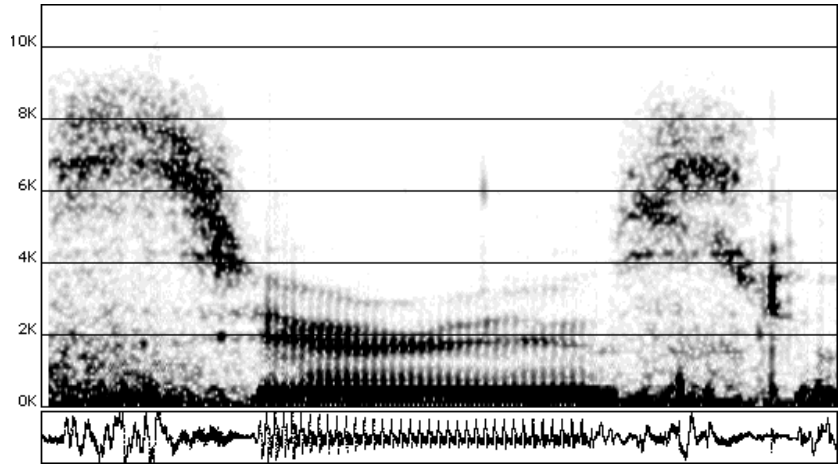

Fig. 1.2. A spectrogram corresponding to the word *serious* [sɪəriəs].

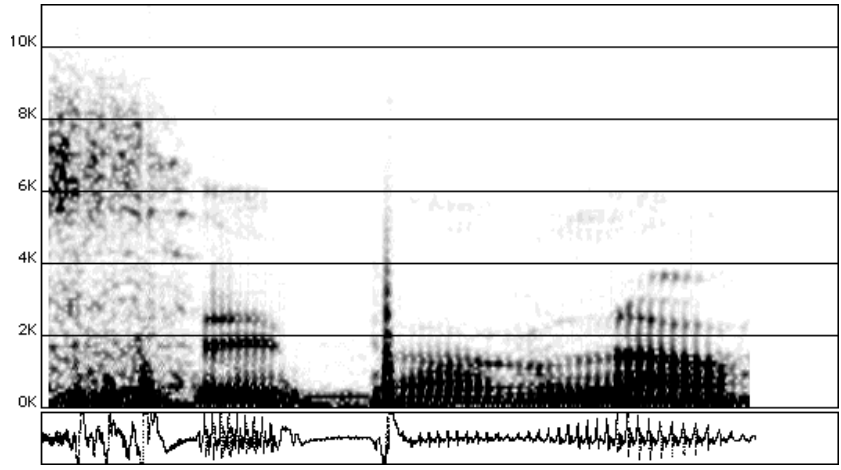

Fig. 1.3. A spectrogram of the French phrase *C'est par là* [separla] 'It is that way'.

Phoneticians can "read" spectrograms, that is, split them into a sequence of relatively regular – stationary – patterns. They can then annotate the corresponding segments with phonemes by recognizing their typical patterns.

A descriptive classification of phonemes includes:

- Simple vowels such as /ɪ/, /a/, and /ɛ/, and nasal vowels in French such as /ɑ̃/ and /ɔ̃/, which appear on the spectrogram as a horizontal bar – the fundamental frequency – and several superimposed horizontal bars – the harmonics.
- Plosives such as /p/ and /b/ that correspond to a stop in the airflow and then a very short and brisk emission of air from the mouth. The air release appears as a vertical bar from 0 to 5,000 Hz.
- Fricatives such as /s/ and /f/ that appear as white noise on the spectrogram, that is, as a uniform gray distribution. Fricatives sounds a bit like a loudspeaker with an unplugged signal cable.
- Nasals and approximants such as /m/, /l/, and /ɾ/ are more difficult to spot and and are subject to modifications according to their left and right neighbors.

Phonemes are assembled to compose words. Pronunciation is basically carried out though **syllables** or diphonemes in European languages. These are more or less stressed or emphasized, and are influenced by neighboring syllables.

The general rhythm of the sentence is the **prosody**. Prosody is quite different from English to French and German and is an open subject of research. It is related to the length and structure of sentences, to questions, and to the meaning of the words.

Speech synthesis uses signal processing techniques, phoneme models, and letter-to-phoneme rules to convert a text into speech and to read it in a loud voice. **Speech recognition** does the reverse and transcribes speech into a computer-readable text. It also uses signal processing and statistical techniques including Hidden Markov models and language models.

1.5 Lexicon and Morphology

The set of available words in a given context makes up a lexicon. It varies from language to language and within a language according to the context: jargon, slang, or gobbledygook. Every word can be classified through a lexical category or **part of speech** such as article, noun, verb, adjective, adverb, conjunction, preposition, or pronoun. Most of the lexical entities come from four categories: noun, verb, adjective, and adverb. Other categories such as articles, pronouns, or conjunctions have a limited and stable number of elements. Words in a sentence can be annotated – tagged – with their part of speech.

For instance, the simple sentences in English, French, and German:

The big cat ate the gray mouse
Le gros chat mange la souris grise
Die große Katze ißt die graue Maus

are annotated as:

The/article *big*/adjective *cat*/noun *ate*/verb *the*/article *gray*/adjective *mouse*/noun
Le/article *gros*/adjectif *chat*/nom *mange*/verbe *la*/article *souris*/nom *grise*/adjectif
Die/Artikel *große*/Adjektiv *Katze*/Substantiv *ißt*/Verb *die*/Artikel *graue*/Adjektiv *Maus*/Substantiv

Morphology is the study of how root words and affixes – the **morphemes** – are composed to form words. Morphology can be divided into **inflection** and **derivation**:

- Inflection is the form variation of a word under certain grammatical conditions. In European languages, these conditions consist notably of the number, gender, conjugation, or tense (Table 1.1).
- Derivation combines affixes to an existing root or stem to form a new word. Derivation is more irregular and complex than inflection. It often results in a change in the part of speech for the derived word (Table 1.2).

Most of the inflectional morphology of words can be described through morphological rules, possibly with a set of exceptions. According to the rules, a morphological parser splits each word as it occurs in a text into morphemes – the root word and the affixes. When affixes have a grammatical content, morphological parsers generally deliver this content instead of the raw affixes (Table 1.3).

Morphological parsing operates on single words and does not consider the surrounding words. Sometimes, the form of a word is ambiguous. For instance, *worked* can be found in *he worked* (*to work* and preterit) or *he has worked* (*to work* and past

Table 1.1. Grammatical features that modify the form of a word.

Features	Values	English	French	German
Number	singular	*a car*	*une voiture*	*ein Auto*
	plural	*two cars*	*deux voitures*	*zwei Autos*
Gender	masculine	*he*	*il*	*er*
	feminine	*she*	*elle*	*sie*
	neuter	*it*		*es*
Conjugation	infinitive	*to work*	*travailler*	*arbeiten*
and	finite	*he works*	*il travaille*	*er arbeitet*
tense	gerund	*working*	*travaillant*	*arbeitend*

Table 1.2. Examples of word derivations.

	Words	Derived words
English	*real*/adjective	*really*/adverb
French	*courage*/noun	*courageux*/adjective
German	*Der Mut*/noun	*mutig*/adjective

Table 1.3. Decomposition of inflected words into a root and affixes.

	Words	Roots and affixes	Lemmas and grammatical interpretations
English	*worked*	*work* + *ed*	*work* + verb + preterit
French	*travaillé*	*travaill* + *é*	*travailler* + verb + past participle
German	*gearbeitet*	*ge* + *arbeit* + *et*	*arbeiten* + verb + past participle

participle). Another processing stage is necessary to remove the ambiguity and to assign (to annotate) each word with a single part-of-speech tag.

A lexicon may simply be a list of all the **inflected** word forms – a wordlist – as they occur in running texts. However, keeping all the forms, for instance, *work*, *works*, *worked*, generates a useless duplication. For this reason, many lexicons retain only a list of canonical words: the **lemmas**. Lemmas correspond to the entries of most ordinary dictionaries. Lexicons generally contain other features, such as the phonetic transcription, part of speech, morphological type, and definition, to facilitate additional processing. Lexicon building involves collecting most of the words of a language or of a domain. It is probably impossible to build an exhaustive dictionary since new words are appearing every day.

Morphological rules enable us to generate all the word forms from a lexicon. Morphological parsers do the reverse operation and retrieve the word root and its affixes from its inflected or derived form in a text. Morphological parsers use finite-state automaton techniques. Part-of-speech taggers disambiguate the possible multiple readings of a word. They also use finite-state automata or statistical techniques.

1.6 Syntax

Syntax governs the formation of a sentence from words. Syntax is sometimes combined with morphology under the term morphosyntax. Syntax has been a central point of interest of linguistics since the Middle Ages, but it probably reached an apex in the 1970s, when it captured an overwhelming attention in the linguistics community.

1.6.1 Syntax as Defined by Noam Chomsky

Chomsky (1957) had a determining influence in the study of language, and his views have fashioned the way syntactic formalisms are taught and used today. Chomsky's theory postulates that syntax is independent from semantics and can be expressed in terms of logic grammars. These grammars consist of a set of rules that describe the sentence structure of a language. In addition, grammar rules can generate the whole sentence set – possibly infinite – of a definite language.

Generative grammars consist of syntactic rules that fractionate a phrase into subphrases and hence describe a sentence composition in terms of phrase structure. Such rules are called **phrase-structure rules**. An English sentence typically comprises

two main phrases: a first one built around a noun called the noun phrase, and a second one around the main verb called the verb phrase. Noun and verb phrases are rewritten into other phrases using other rules and by a set of terminal symbols representing the words.

Formally, a grammar describing a very restricted subset of English, French, or German phrases could be the following rule set:

- A **sentence** consists of a **noun phrase** and a **verb phrase**.
- A **noun phrase** consists of an **article** and a **noun**.
- A **verb phrase** consists of a **verb** and a **noun phrase**.

A very limited lexicon of the English, French, or German words could be made of:

- articles such as *the*, *le*, *la*, *der*, *den*
- nouns such as *boy*, *garçon*, *Knabe*
- verbs such as *hit*, *frappe*, *trifft*

This grammar generates sentences such as:

The boy hit the ball
Le garçon frappe la balle
Der Knabe trifft den Ball

but also incorrect or implausible sequences such as:

The ball hit the ball
**Le balle frappe la garçon*
**Das Ball trifft den Knabe*

Linguists use an asterisk (*) to indicate an ill-formed grammatical construction or a nonexistent word. In the French and German sentences, the articles must agree with their nouns in gender, number, and case (for German). The correct sentences are:

La balle frappe le garçon
Der Ball trifft den Knaben

Trees can represent the syntactic structure of sentences (Fig. 1.4–1.6) and reflect the rules involved in sentence generation.

Moreover, Chomsky's formalism enables some transformations: rules can be set to carry out the building of an interrogative sentence from a declaration, or the building of a passive form from an active one.

Parsing is the reverse of generation. A grammar, a set of phrase-structure rules, accepts syntactically correct sentences and determines their structure. Parsing requires a mechanism to search the rules that describe the sentence's structure. This mechanism can be applied from the sentence's words up to a rule describing the sentence's structure. This is **bottom-up parsing**. Rules can also be searched from a sentence structure rule down to the sentence's words. This corresponds to **top-down parsing**.

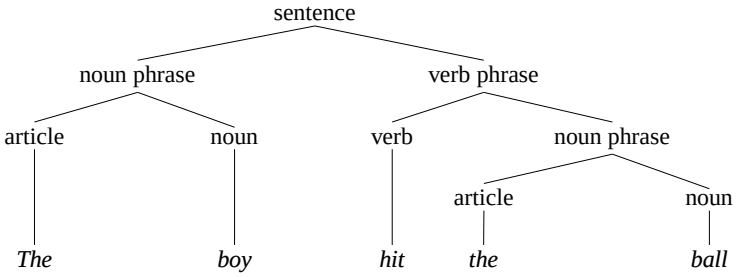

Fig. 1.4. Tree structure of *The boy hit the ball*.

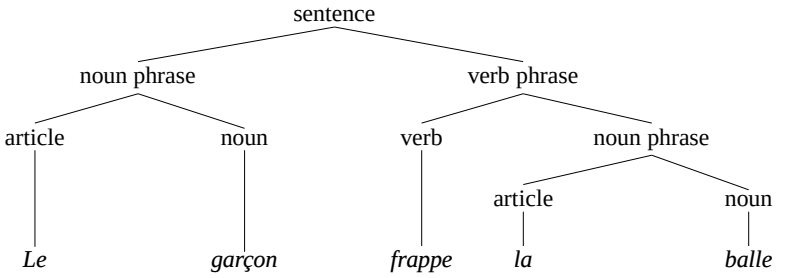

Fig. 1.5. Tree structure of *Le garçon frappe la balle*.

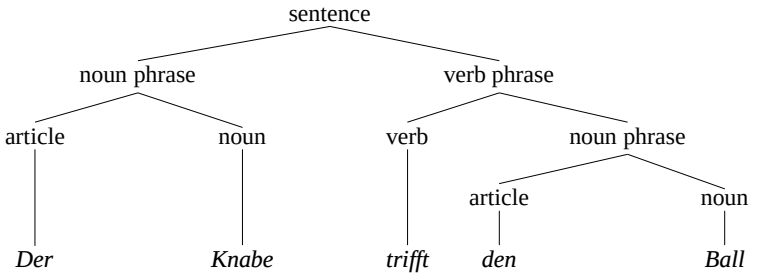

Fig. 1.6. Tree structure of *Der Knabe trifft den Ball*.

1.6.2 Syntax as Relations and Dependencies

Before Chomsky, pupils and students learned syntax (and still do so) mainly in terms of functions and relations between the words. A sentence's classical parsing consists in annotating words using parts of speech and in identifying the main verb. The main verb is the pivot of the sentence, and the principal grammatical functions are determined relative to it. Parsing consists then in grouping words to form the subject and the object, which are the two most significant functions in addition to the verb.

In the sentence *The boy hit the ball*, the main verb is *hit*, the subject of *hit* is *the boy*, and its object is *the ball* (Fig. 1.7).

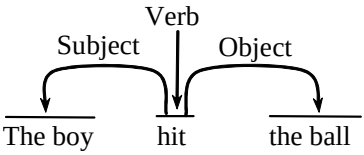

Fig. 1.7. Grammatical relations in the sentence *The boy hit the ball*.

Other grammatical functions (or relations) involve notably articles, adjectives, and adjuncts. We see this in the sentence

The big boy from Liverpool hit the ball with furor.

where the adjective *big* is related to the noun *boy*, and the adjuncts *from Liverpool* and *with furor* are related respectively to *boy* and *hit*.

We can picture these relations as a dependency net, where each word is said to modify exactly another word up to the main verb (Fig. 1.8). The main verb is the head of the sentence and modifies no other word. Tesnière (1966) and Mel'cuk (1988) have extensively described dependency theory.

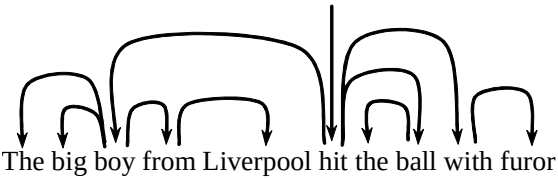

Fig. 1.8. Dependency relations in the sentence *The big boy from Liverpool hit the ball with furor*.

Although they are less popular than phrase-structure grammars, **dependency grammars** often prove more efficient to parse texts. They provide a theoretical framework to many present parsing techniques and have numerous applications.

1.7 Semantics

The semantic level is more difficult to capture and there are numerous viewpoints on how to define and to process it. A possible viewpoint is to oppose it to syntax: there are sentences that are syntactically correct but that cannot make sense. Such a description of semantics would encompass sentences that make sense. Classical examples by Chomsky (1957) – sentences 1 and 2 – and Tesnière (1966) – sentence 3 – include:

1. *Colorless green ideas sleep furiously.*
2. **Furiously sleep ideas green colorless.*
3. *Le silence vertébral indispose la voile licite.*
   'The vertebral silence embarrasses the licit sail.'

Sentences 1 and 3 and are syntactically correct but have no meaning, while sentence 2 is neither syntactically nor semantically correct.

In computational linguistics, semantics is often related to logic and to predicate calculus. Determining the semantic representation of a sentence then involves turning it into a predicate-argument structure, where the predicate is the main verb and the arguments correspond to phrases accompanying the verb such as the subject and the object. This type of logical representation is called a **logical form**. Table 1.4 shows examples of sentences together with their logical forms.

Table 1.4. Correspondence between sentences and logical forms.

Sentences	**Logical forms (predicates)**
Pierre wrote notes	`wrote(pierre, notes).`
Pierre a écrit des notes	`a_écrit(pierre, notes).`
Pierre schrieb Notizen	`schrieb(pierre, notizen).`

Representation is only one facet of semantics. Once sentence representations have been built, they can be interpreted to check what they mean. *Notes* in the sentence *Pierre wrote notes* can be linked to a dictionary **definition**. If we look up in the *Cambridge International Dictionary of English* (Procter 1995), there are as many as five possible senses for *notes* (abridged from p. 963):

1. **note** [WRITING], *noun*, a short piece of writing;
2. **note** [SOUND], *noun*, a single sound at a particular level;
3. **note** [MONEY], *noun*, a piece of paper money;
4. **note** [NOTICE], *verb*, to take notice of;
5. **note** [IMPORTANCE], *noun*, of note: of importance.

So linking a word meaning to a definition is not straightforward because of possible ambiguities. Among these definitions, the intended sense of *notes* is a specialization of the first entry:

notes, *plural noun*, notes are written information.

Finally, *notes* can be interpreted as what they refer to concretely, that is, a specific object: a set of bound paper sheets with written text on them or a file on a computer disk that keeps track of a set of magnetic blocks. Linking a word to an object of the real world, here a file on a computer, is a part of semantics called **reference resolution**.

The **referent** of the word *notes*, that is, the designated object, could be the path `/users/pierre/language_processing.html` in Unix parlance. As

for the definition of a word, the referent can be ambiguous. Let us suppose that a database contains the locations of the lecture notes Pierre wrote. In Prolog, listing its content could yield:

```
notes('/users/pierre/operating_systems.html').
notes('/users/pierre/language_processing.html').
notes('/users/pierre/prolog_programming.html').
```

Here this would mean that finding the referent of *notes* consists in choosing a document among three possible ones (Fig. 1.9).

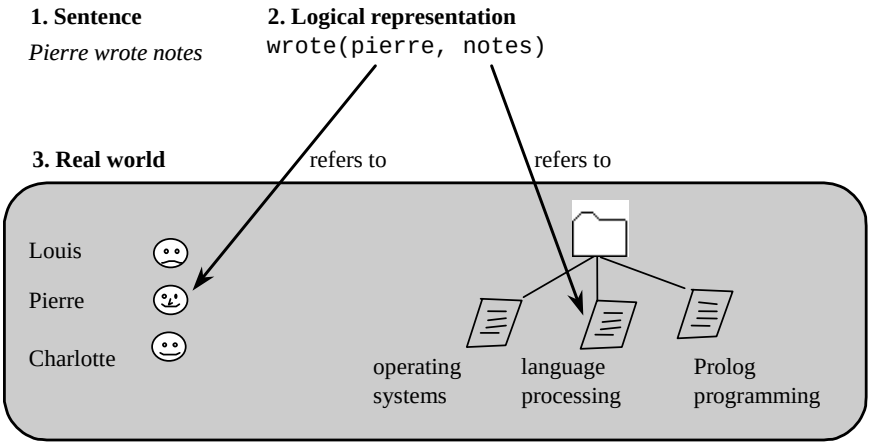

Fig. 1.9. Resolving references of *Pierre wrote notes*.

Obtaining the semantic structure of a sentence has been discussed abundantly in the literature. This is not surprising, given the uncertain nature of semantics. Building a logical form often calls on the **composition** of the semantic representation of the phrases that constitute a sentence. To carry it out, we must assume that sentences and phrases have an internal representation that can be expressed in terms of a logical formula.

Once a representation has been built, a reasoning process is applied to resolve references and to determine whether a sentence is true or not. It generally involves rules of deduction, or **inferences**.

Pragmatics is semantics restricted to a specific context and relies on facts that are external to the sentence. These facts contribute to the inference of a sentence's meaning or prove its truth or falsity. For instance, pragmatics of

Methuselah lived to be 969 years old. (Genesis 5:27)

can make sense in the Bible but not elsewhere, given the current possibilities of medicine.

1.8 Discourse and Dialogue

An interactive conversational agent cannot be envisioned without considering the whole **discourse** of (human) users – or parts of it – and apart from a **dialogue** between a user and the agent. Discourse refers to a sequence of sentences, to a sentence context in relation with other sentences or with some background situation. It is often linked with pragmatics.

Discourse study also enables us to resolve references that are not self-explainable in single sentences. Pronouns are good examples of such missing information. In the sentence

John took it

the pronoun *it* can probably be related to an entity mentioned in a previous sentence, or is obvious given the context where this sentence was said. These references are given the name of **anaphors**.

Dialogue provides a means of communication. It is the result of two intermingled – and, we hope, interacting – discourses: one from the user and the other from the machine. It enables a conversation between the two entities, the assertion of new results, and the cooperative search for solutions.

Dialogue is also a tool to repair communication failures or to complete interactively missing data. It may clarify information and mitigate misunderstandings that impair communication. Through a dialogue a computer can respond and ask the user:

I didn't understood what you said! Can you repeat (rephrase)?

Dialogue easily replaces some hazardous guesses. When an agent has to find the potential reference of a pronoun or to solve reference ambiguities, the best option is simply to ask the user clarify what s/he means:

Tracy? Do you mean James' brother or your mother?

Discourse processing splits texts and sentences into segments. It then sets links between segments to chain them rationally and to map them onto a sort of structure of the text. Discourse studies often make use of **rhetoric** as a background model of this structure.

Dialogue processing classifies the segments into what are called **speech acts**. At a first level, speech acts comprise dialogue turns: the user turn and the system turn. Then turns are split into sentences, and sentences into questions, declarations, requests, answers, etc. Speech acts can be modeled using finite-state automata or more elaborate schemes using **intention** and **planning** theories.

1.9 Why Speech and Language Processing Are Difficult

For all the linguistic levels mentioned in the previous sections, we outlined models and techniques to process speech and language. They often enable us to obtain excellent results compared to the performance of human beings. However, for most levels,

language processing rarely hits the ideal score of 100%. Among the hurdles that often prevent the machine from reaching this figure, two recur at any level: ambiguity and the absence of a perfect model.

1.9.1 Ambiguity

Ambiguity is a major obstacle in language processing, and it may be the most significant. Although as human beings we are not aware of it most of the time, ambiguity is ubiquitous in language and plagues any stage of automated analysis. We saw examples of ambiguous morphological analysis and part-of-speech annotation, word senses, and references. Ambiguity also occurs in speech recognition, parsing, anaphora solving, and dialogue.

McMahon and Smith (1996) illustrate strikingly ambiguity in speech recognition with the sentence

The boys eat the sandwiches.

Speech recognition comprises generally two stages: first, a phoneme recognition, and then a concatenation of phoneme substrings into words. Using the International Phonetic Association (IPA) symbols, a perfect phonemic transcription of this utterance would yield the transcription:

[ˈðəbˈoɪzˈiːtˈðəsˈændwɪdʒɪz],

which shows eight other alternative readings at the word decoding stage:

**The boy seat the sandwiches.*
**The boy seat this and which is.*
**The boys eat this and which is.*
The buoys eat the sandwiches.
**The buoys eat this and which is.*
The boys eat the sand which is.
**The buoys seat this and which is.*

This includes the strange sentence

The buoys eat the sand which is.

For syntactic and semantic layers, a broad classification occurs between lexical and structural ambiguity. Lexical ambiguity refers to multiple senses of words, while structural ambiguity describes a parsing alternative, as with the frequently quoted sentence

I saw the boy with a telescope,

which can mean either that I used a telescope to see the boy or that I saw the boy who had a telescope.

A way to resolve ambiguity is to use a conjunction of language processing components and techniques. In the example given by McMahon and Smith, five out of

eight possible interpretations are not grammatical. These are flagged with an asterisk. A further syntactic analysis could discard them.

Probabilistic models of word sequences can also address disambiguation. Statistics on word occurrences drawn from large quantities of texts – corpora – can capture grammatical as well as semantic patterns. Improbable alternatives <boys eat sand> and <buoys eat sand> are also highly unlikely in corpora and will not be retained (McMahon and Smith 1996). In the same vein, probabilistic parsing is a very powerful tool to rank alternative parse trees, that is, to retain the most probable and reject the others.

In some applications, logical rules model the context, reflect common sense, and discard impossible configurations. Knowing the physical context may help disambiguate some structures, as in the boy and the telescope, where both interpretations of the isolated sentence are correct and reasonable. Finally, when a machine interacts with a user, it can ask her/him to clarify an ambiguous utterance or situation.

1.9.2 Models and Their Implementation

Processing a linguistic phenomenon or layer starts with the choice or the development of a formal model and its algorithmic implementation. In any scientific discipline, good models are difficult to design. This is specifically the case with language. Language is closely tied to human thought and understanding, and in some instances models in computational linguistics also involve the study of the human mind. This gives a measure of the complexity of the description and the representation of language.

As noted in the introduction, linguists have produced many theories and models. Unfortunately, few of them have been elaborate enough to encompass and describe language effectively. Some models have also been misleading. This explains somewhat the failures of early attempts in language processing. In addition, many of the potential theories require massive computing power. Processors and storage able to support the implementation of complex models with substantial dictionaries, corpora, and parsers were not widely available until recently.

However, in the last decade models have matured, and computing power has become inexpensive. Although models and implementations are rarely (never?) perfect, they now enable us to obtain exploitable results. Most use a limited set of techniques that we will consider throughout this book, namely finite-state automata, logic grammars, and first-order logic. These tools are easily implemented in Prolog. Another set of tools pertains to the theory of probability and statistics. The combination of logic and statistical techniques now enables us to parse running-text sentences with a success rate of nearly 90%, a figure that would have been unimaginable ten years ago.

1.10 An Example of Language Technology in Action: the Persona Project

1.10.1 Overview of Persona

The Persona prototype from Microsoft Research (Ball et al. 1997) illustrates a user interface that is based on a variety of language processing techniques. Persona is a conversational agent that helps a user select songs and music tracks from a record database. Peedy, an animated cartoonlike parrot, embodies the agent that interacts with the user. It contains speech recognition, parsing, and semantic analysis modules to listen and to respond to the user and to play the songs. Table 1.5 shows an example of a dialogue with Peedy.

Table 1.5. An excerpt of a Persona dialogue. After Ball et al. (1997).

Turns	Utterances
	[Peedy is asleep on his perch]
User:	Good morning, Peedy.
	[Peedy rouses]
Peedy:	**Good morning.**
User:	Let's do a demo.
	[Peedy stands up, smiles]
Peedy:	**Your wish is my command, what would you like to hear?**
User:	What have you got by Bonnie Raitt?
	[Peedy waves in a stream of notes, and grabs one as they rush by.]
Peedy:	**I have "The Bonnie Raitt Collection" from 1990.**
User:	Pick something from that.
Peedy:	**How about "Angel from Montgomery"?**
User:	Sounds good.
	[Peedy drops note on pile]
Peedy:	**OK.**
User:	Play some rock after that.
	[Peedy scans the notes again, selects one]
Peedy:	**How about "Fools in Love"?**
User:	Who wrote that?
	[Peedy cups one wing to his 'ear']
Peedy:	**Huh?**
User:	Who wrote that?
	[Peedy looks up, scrunches his brow]
Peedy:	**Joe Jackson**
User:	Fine.
	[Drops note on pile]
Peedy:	**OK.**

Certain interactive talking assistants consider a limited set of the linguistic levels we have presented before. Simple systems bypass syntax, for example, and have

only a speech recognition device to detect a couple of key words. In contrast, Persona has components to process more layers. They are organized in modules carrying out speech recognition, speech synthesis, parsing, semantics analysis, and dialogue. In addition, Persona has components specific to the application such as a name substitution module to find proper names like *Madonna* or *Debussy* and an animation module to play the Peedy character.

Persona's architecture organizes its modules into a pipeline processing flow (Fig. 1.10). Many other instances of dialogue systems adopt a similar architecture.

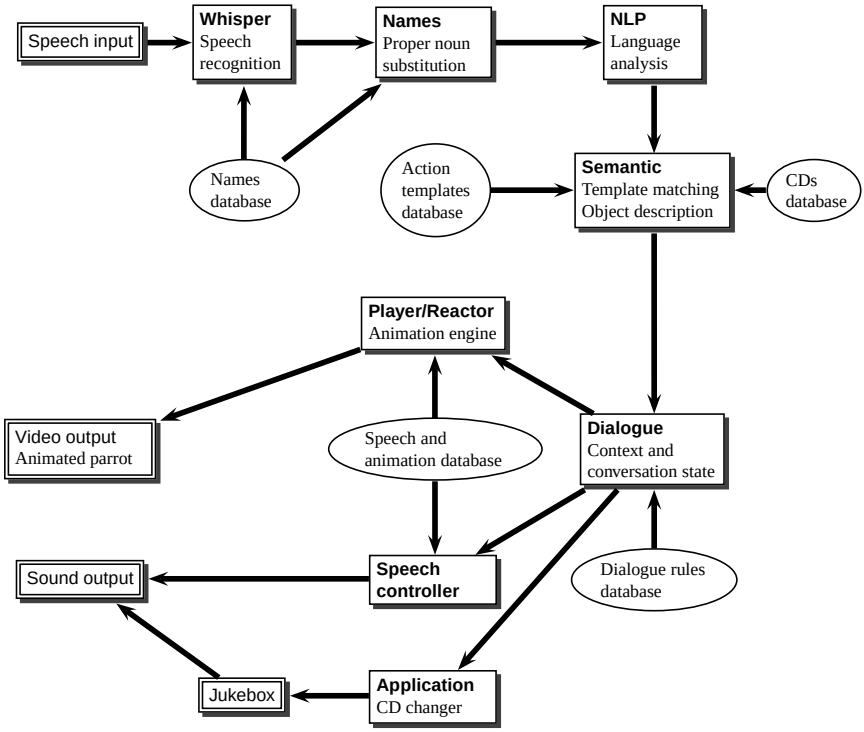

Fig. 1.10. Architecture of the Persona conversational assistant. After Ball et al. (1997).

1.10.2 The Persona's Modules

Persona's first component is the Whisper speech recognition module (Huang et al. 1995). Whisper uses signal processing techniques to compare phoneme models to the acoustic waves, and it assembles the recognized phonemes into words. It also uses a grammar to constrain the recognition possibilities. Whisper transcribes continuous speech into a stream of words in real time. It is a speaker-independent system. This means that it operates with any speaker without training.

The user's orders to select music often contain names: artists, titles of songs, or titles of albums. The Names module extracts them from the text before they are passed on to further analysis. Names uses a pattern matcher that attempts to substitute all the names and titles contained in the input sentence with placeholders. The utterance *Play before you accuse me by Clapton* is transformed into *Play* `track1` *by* `artist1`.

The NLP module parses the input in which names have been substituted. It uses a grammar with rules similar to that of Sect. 1.6.1 and produces a tree structure. It creates a logical form whose predicate is the verb and the arguments the subject and the object: `verb(subject, object)`. The sentence *I would like to hear something* is transformed into the form `like(i, hear(i, something))`.

The logical forms are converted into a task graph representing the utterance in terms of actions the agent can do and objects of the task domain. It uses an application-dependent notation to map English words to symbols. It also reverses the viewpoint from the user to the agent. The logical form of *I would like to hear something* is transformed into the task graph: `verbPlay(you, objectTrack)` – *You play (*`verbPlay`*) a track (*`objectTrack`*).*

Each possible request Peedy understands has possible variations – paraphrases. The mapping of logical forms to task graphs uses transformation rules to reduce them to a limited set of 17 canonical requests. The transformation rules deal with synonyms, syntactic variation, and colloquialisms. The forms corresponding to

I'd like to hear some Madonna.
I want to hear some Madonna.
It would be nice to hear some Madonna.

are transformed into a form equivalent to

Let me hear some Madonna.

The resulting graph is matched against actions templates the jukebox can carry out.

The dialogue module controls Peedy's answers and reactions. It consists of a state machine that models a sequence of interactions. Depending on the state of the conversation and an input event – what the user says – Peedy will react: trigger an animation, utter a spoken sentence or play music, and move to another conversational state.

1.11 Further Reading

Introductory textbooks to linguistics include *An Introduction to Language* (Fromkin et al. 2003) and *Linguistics: An Introduction to Linguistics Theory* (Fromkin 2000). *Linguistics: The Cambridge Survey* (Newmeyer et al. 1988) is an older reference in four volumes. The *Nouveau dictionnaire encyclopédique des sciences du langage* (Ducrot and Schaeffer 1995) is an encyclopedic presentation of linguistics in French,

and *Studienbuch Linguistik* (Linke et al. 2004) is an introduction in German. *Fundamenti di linguistica* (Simone 1998) is an outstandingly clear and concise work in Italian that describes most fundamental concepts of linguistics.

Concepts and theories in linguistics evolved continuously from their origins to the present time. Historical perspectives are useful to understand the development of central issues. *A Short History of Linguistics* (Robins 1997) is a very readable introduction to linguistics history. *Histoire de la linguistique de Sumer à Saussure* (Malmberg 1991) and *Analyse du langage au XX^e siècle* (Malmberg 1983) are comprehensive and accessible books that review linguistic theories from the ancient Near East to the end of the 20th century. *Landmarks in Linguistic Thought, The Western Tradition from Socrates to Saussure* (Harris and Taylor 1997) are extracts of founding classical texts followed by a commentary.

The journal of best repute in the domain of computational linguistics is *Computational Linguistics*, published by the Association for Computational Linguistics (ACL). Some interesting articles can also be found in the ACL conference proceedings and in more general journals such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, other IEEE journals, *Artificial Intelligence*, and the Association for Computing Machinery (ACM) journals. The French journal *Traitement automatique des langues* is also a source of interesting papers. It is published by the Association de traitement automatique des langues (http://www.atala.org).

Available books on natural language processing include (in English): *Natural Language Processing in Prolog* (Gazdar and Mellish 1989), *Prolog for Natural Language Analysis* (Gal et al. 1991), *Natural Language Processing for Prolog Programmers* (Covington 1994), *Natural Language Understanding* (Allen 1994), *Foundations of Statistical Natural Language Processing* (Manning and Schütze 1999), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Jurafsky and Martin 2000), *Foundations of Computational Linguistics: Human-Computer Communication in Natural Language* (Hausser 2001). Avalaible books in French include: *Prolog pour l'analyse du langage naturel* (Gal et al. 1989), *L'intelligence artificielle et le langage* (Sabah 1990), and in German *Grundlagen der Computerlinguistik. Mensch-Maschine-Kommunikation in natürlicher Sprache* (Hausser 2000).

There are plenty of interesting resources on the Internet. Web sites include digital libraries, general references, corpus and lexical resources, together with software registries. A starting point is the official home page of the ACL, which provides many links (http://www.aclweb.org). An extremely valuable anthology of papers published under the auspices of the ACL is available from this site (http://www.aclweb.org/anthology). Wikipedia (http://www.wikipedia.org) is a free encyclopedia that contains definitions and general articles on concepts and theories used in computational linguistics and natural language processing.

Many source programs are available on the Internet, either free or under a license. They include speech synthesis and recognition, morphological analysis, parsing, and so on. The German Institute for Artificial Intelligence Research maintains a list of them at the Natural Language Software Registry (http://registry.dfki.de).

Lexical and corpus resources are now available in many languages. Valuable sites include the Oxford Text Archive (http://ota.ox.ac.uk/), the Linguistic Data Consortium of the University of Pennsylvania (http://www.ldc.upenn.edu/), and the European Language Resources Association (http://www.elra.info).

There are nice interactive online demonstrations covering speech synthesis, parsing, translation and so on. Since sites are sometimes transient, we don't list them here. A good way to find them is to use directories like Yahoo, or search engines like Google.

Finally, some companies and laboratories have a very active research in language processing. They include major software powerhouses like Microsoft, IBM, and Xerox. The paper describing the Peedy animated character can be found at the Microsoft Research Web site (http://www.research.microsoft.com).

Exercises

1.1. List some computer applications that are relevant to the domain of language processing.

1.2. Tag the following sentences using parts of speech you know:
The cat caught the mouse.
Le chat attrape la souris.
Die Katze fängt die Maus.

1.3. Give the morpheme list of: *sings*, *sung*, *chante*, *chantiez*, *singt*, *sang*. List all the possible ambiguities.

1.4. Give the morpheme list of: *unpleasant*, *déplaisant*, *unangenehm*.

1.5. Draw the tree structure of the sentences:
The cat caught the mouse.
Le chat attrape la souris.
Die Katze fängt die Maus.

1.6. Identify the main functions of these sentences and draw the corresponding dependency net linking the words:
The cat caught the mouse.
Le chat attrape la souris.
Die Katze fängt die Maus.

1.7. Draw the dependency net of the sentences:
The mean cat caught the gray mouse on the table.
Le chat méchant a attrapé la souris grise sur la table.
Die böse Katze hat die graue Maus auf dem Tisch gefangen.

1.8. Give examples of sentences that are:
- Syntactically incorrect
- Syntactically correct
- Syntactically and semantically correct

1.9. Give the logical form of these sentences:
The cat catches the mouse.
Le chat attrape la souris.
Die Katze fängt die Maus.

1.10. Find possible phonetic interpretations of the French phrase *quant-à-soi*.

1.11. List the components you think necessary to build a spoken dialogue system.