Games, walks and grammars: Problems I've worked on

Vervoort, M.R.

Publication date
2000

Chapter 11

The Basics of EMILE

11.1 Introduction

Human being are remarkably good in working with natural languages. Even if someone has no knowledge of the formal structure of a language, he or she will be able to tell when 'something' is like 'something else'. For instance, Lewis Caroll's famous poem 'Jabberwocky' starts with

> 'Twas brillig, and the slithy toves
> Did gyre and gimble in the wabe:
> All mimsy were the borogoves
> and the mome raths outgrabe.

Even without Humpty Dumpty's annotations, it is immediately obvious what the *syntactic structure* of the first sentence is: 'brillig' and 'slithy' are adjectives, 'toves' is a noun, 'gyre' and 'gimble' are verbs, etcetera.
So how do we know such things? The short answer is 'from context'. When a sentence starts with ' 'Twas', we are not surprised if the next word is an adjective. Similarly, if a sentence has the pattern 'the (.) did (.) and (.) in the (.)', we expect the missing phrases to be a noun-phrase, two verb-phrases and another noun-phrase, respectively.
The notion of *grammatical type* has many possible definitions. For instance, if we had a *context-free* grammar of a language, we can view each non-terminal symbol as a grammatical type. In general, one of the properties of a grammatical type is, that wherever some expression is used as an expression of that type, other expressions of that type can be substituted without making the sentence ungrammatical. This gives rise to a notion of a grammatical type as a set of expressions together with a set of contexts. For instance, the type 'noun-phrase' could be represented by the set of all noun-phrases, together with the set of all contexts in which a noun-phrase can appear. Combining any of the expressions of

a type with any of the contexts will yield a grammatical sentence. Many of these combinations might appear in actual texts, especially the short ones (in terms of number of words).

In this terminology, we can describe the above phenomenon, as the existence of contexts which are *characteristic* for a type, meaning that whenever something appears in that context, we assume it also belongs to that type. Some types may also have *characteristic expressions*, with the analogous property.

EMILE[30] 4.1 is a program based on the above concepts. It attempts to learn the grammatical structure of a language from sentences of that language, without being given any prior knowledge of the grammar. For any type in any valid grammar for the language, we can expect context/expression combinations to show up in a sufficiently large sample of sentences of the language. EMILE searches for such clusters of expressions and contexts in the sample, and interprets them as grammatical types. It then tries to find characteristic contexts and expressions, and uses them to extend the types. Finally, it formulates derivation rules based on the types found, in the manner of the rules of a context-free grammar. The program can present the grammatical structure found in several ways, as well as use it to parse other sentences or generate new ones.

The theoretical concepts used in EMILE 4.1 are elaborated on in P. Adriaans articles on EMILE 1.0/2.0 [33] and EMILE 3.0 [34], in these chapters we will focus on the practical aspects. Note that although EMILE 4.1 is based on the same theoretical concepts as EMILE 3.0, it is not based on the same algorithm. More information on the precursors of EMILE 4.1 may be found in the above articles, as well as in the E. Dörnenburg's Master's Thesis[36].

11.2 Definitions

The three most basic concepts in EMILE are *contexts*, *expressions* and *context/expression pairs*.

11.2.1. Definition. A *context/expression pair* is a sentence split into three parts, for instance

John (makes) tea

Here, 'makes' is called an *expression*, and 'John (.) tea' is called a *context* (with *left-hand side* 'John' and *right-hand side* 'tea').

11.2.2. Definition. We say that an expression e *appears with* a context c, or that the context/expression pair (c, e) has been *encountered*, if $c_l\hat{}e\hat{}c_r$ appears

[30] EMILE 4.1 is a successor to EMILE 3.0, written by P. Adriaans. The original acronym stands for Entity Modeling Intelligent Learning Engine. It refers to earlier versions of EMILE that also had semantic capacities. The name EMILE is also motivated by the book on education by J.-J. Rousseau.

as a sentence in a text, where c_l and c_r are the left-hand side and the right-hand side of c, respectively, and $a^\frown b$ denotes the concatenation of a and b.

11.2.3. REMARK. Context/expression pairs are not always sensible, as for instance in the sentence

John (drinks coffee, and Mary drinks) tea

where the expression 'drinks coffee, and Mary drinks' appears in the context 'John (.) tea'. EMILE will find such context/expression pairs and attempt to use them in the grammar induction process. But such pairs are usually isolated, i.e. they are not part of any significant clusters. So EMILE will fail to make use of them, and they will be effectively ignored.

As stated before, we view grammatical types in terms of the expressions that belong to that type, and the contexts in which they can appear (as expressions of that type). As such, we define grammatical types as follows:

11.2.4. DEFINITION. In the context of this paper, a grammatical type T is defined as a pair (T_C, T_E), where T_C is a set of contexts, and T_E is a set of expressions. Elements of T_C and T_E are called *primary* contexts and expressions for T.

The intended meaning of this definition is, that all expressions of a type can appear with all of its the contexts.
In natural languages, the type of an expression is not always unambiguous. For instance, the word 'walk' can be both a noun and a verb. Hence 'walk' will not only appear in contexts for noun-phrases, but also in contexts for verb-phrases. The same does not hold for the phrase 'thing': 'thing' only appears in contexts for noun-phrases, and in any such context, any noun can be substituted for 'thing' without making the sentence ungrammatical. We say that 'thing' is *characteristic* for the type 'noun'. Formally,

11.2.5. DEFINITION. An expression of a type T is *characteristic* for T if it only appears with contexts of type T.[31] Similarly, a context of a type T is *characteristic* for T if it only appears with expressions of type T.

In these chapters, we will also use *characteristic** and *secondary* expressions and contexts. However, as these definitions are rather dependent on the algorithms, they will be delayed until section 12.4. That section also has several examples of characteristic expressions and contexts.

[31] EMILE changes this definition slightly in implementation, in that contexts which have been assigned no type at all are completely ignored, i.e. an expression is characteristic if all contexts with which it appears are of type T, or untyped.

NOTATION. For any type T, T_E, T_E^{ch}, T_E^* and T_C^{se} denote the sets of primary, characteristic, characteristic* and secondary expressions of T, and T_C, T_C^{ch}, T_C^* and T_C^{se} denote the corresponding sets of contexts.

The EMILE program also attempts to transform the collection of grammatical types found into a context-free grammar consisting of derivation rules. Such rules generally are of the form

$$[T] \Rightarrow s_0[T_1]s_1[T_2]\ldots[T_k]s_k$$

where $T, T_1, T_2, \ldots, T_k$ are grammatical types, and $s_0, s_1, \ldots, s_k$ are (possibly empty) sequences of words. Given a rule with left-hand side $[T]$, and a sequence of word-sequences and grammatical types containing $[T]$, that appearance of $[T]$ can be replaced by the right-hand side of the rule, (concatenating adjacent word-sequences as necessary). Any sequence which can be obtained from another sequence by such rule applications, is said to be derivable from that sequence. The language of a context-free grammar consists of those word-sequences e such that $[0] \Rightarrow e$ is derivable, where $[0]$ denotes the type of whole sentences.