MACHINE LEARNING OF MORPHOLOGICAL RULES BY GENERALIZATION AND ANALOGY

Klaus Wothke
Arbeitsstelle Linguistische Datenverarbeitung
INSTITUT FÜR DEUTSCHE SPRACHE
Mannheim, West Germany

ABSTRACT: This paper describes an experimental procedure for the inductive automated learning of morphological rules from examples. At first an outline of the problem is given. Then a formalism for the representation of morphological rules is defined. This formalism is used by the automated procedure, whose anatomy is subsequently presented. Finally the performance of the system is evaluated and the most important unsolved problems are discussed.

1. Outline of the Problem

Learning algorithms for the domain of natural languages were in the past mainly developed to model the acquisition of syntax and to generate syntactic descriptions from examples (cf. Pinker 1979, Cohen/Feigenbaum 1982: 494-511). There exist also some systems which learn rules for the automatic phonetic transcription of orthographic text (cf. Oakey/Cawthorn 1981, Wolf 1977). Like the system presented in this paper all these systems still are experimental systems. The inductive automatic learning of morphological rules has till now been investigated only to a small degree. Research on this problem was carried out by Ring (1978), Jansen-Winkeln (1985) and Wothke (1985).

The task of the system described here is to learn rules for inflectional and derivational morphology. The system is not designed as a standard program, but as an experimental system. It is used for the experimental development and the testing of fundamental algorithmic learning strategies. Later these strategies could perhaps become necessary components of a standard learning program devised for the interactive development of linguistic algorithms for the domain of morphology.

Input to the system is a set of examples called a learning corpus. Each example is an ordered pair of words. We call the first word of each pair the source. The second word is called the target. Between the source and the target of each given pair there must exist an inflectional or a derivational morphological relation. By applying the processes of generalization and detection analogies the system has to construct a set of instructions which describe on a purely graphemic basis how the target of each pair is generated from the source. (Semantic features of morphemes are at present ignored by the system.) Such a set of instructions should not only generate correct targets for the sources given in the learning corpus: The instructions should also generate correct targets for the majority of the sources not in the corpus which participate in the same inflectional or derivational relationship as the source-target-pairs in the learning corpus. Suppose for example that the following learning corpus is fed into the system:

´assembly´	´assemblies´
´bath´	´baths´
´box´	´boxes´
´boy´	´boys´
´bus´	´buses´
´bush´	´bushes´
´buzz´	´buzzes´
´calf´	´calves´
´copy´	´copies´
´cry´	´cries´
´door´	´doors´
´field´	´fields´
´house´	´houses´
´knife´	´knives´
´lady´	´ladies´
´mother´	´mothers´
´switch´	´switches´
´university´	´universities´

Figure 1.

In this case the learning algorithm has to construct a set of instructions which generates for each singular noun (= source, in the left column) of this corpus a string which is identical with the corresponding plural form (= target, in the right column). Furthermore, the instructions should also generate the correct plural form for the majority of English singular nouns which are not members of the learning corpus. For instance, the instructions should also generate ´flies´ from ´fly´, ´tables´ from ´table´, ´foxes´ from ´fox´, ´toys´ from ´toy´, ´classes´ from ´class´, and ´thieves´ from ´thief´. Of course there will also be singular nouns for which the instructions will not be adequate. These will include all nouns whose pattern of pluralization is not represented by examples in the learning corpus. With the given learning corpus one could not expect the inferred instructions to be adequate e. g. for the pluralizations ´ox´ -> ´oxen´, ´tooth´ -> ´teeth´, ´index´ -> ´indices´, ´foot´ -> ´feet´, and ´addendum´ -> ´addenda´. As this example illustrates, the linguistic adequacy of the instructions does not only depend on the quality of the automated learning strategies but also on the representativity of a given learning corpus for a morphological pattern.

2. Formalism for the Representation of Morphological Rules

There are two main types of instruction the learning algorithm uses for the formulation of morphological rules:

- Prefixal substitution instructions change the beginning of a source in order to generate the corresponding target. They have the general form

  ```
  X -> Y/#__(Z(1)| ... |Z(i)| ... |Z(n)).
  ```

  Such an instruction means: If a source begins with the string X and if immediately on the right of X follows the string Z(1) or ... or Z(i) or ... or Z(n), then substitute X by Y. (´#´ signifies the word-boundary and ´__´ marks the position where X must occur in order to be substitutable by Y, namely at the beginning of a source (right of ´#´) and immediately before Z(1) or ... or Z(i) or ... or Z(n)).
- Suffixal substitution instructions change the end of a source in order to generate the corresponding target. They have the form

  ```
  X -> Y/(Z(1)| ... |Z(i)| ... |Z(n))__#.
  ```

  The meaning of such an instruction is:If a source ends with the string X and if immediately on the left of X is the string Z(1) or ... or Z(i) or ...or Z(n), then substitute X by Y.

Each set of instructions constructed by the learning algorithm is ordered, i. e. the later application of the instructions to a given source must be tried in a fixed sequence in order to generate a target: The first applicable prefixal instruction in the sequence of prefixal substitution instructions must be determined and the first applicable suffixal instruction in the sequence of suffixal substitution instructions must be determined. Then, both must be applied to the source concurrently, thus generating the target.

The order and application of sets of instructions may be illustrated by a small example: Suppose the learning algorithm has constructed the following set of instructions for the negation of English adjectives (the set is linguistically not fully adequate; ´´ is the nullstring, i. e. the string with the length 0):

```
(1) ´´ -> ´il´/#__´l´
(2) ´´ -> ´ir´/#__´r´
(3) ´´ -> ´im´/#__(´m´|´p´)
(4) ´´ -> ´in´/#__

(5) ´´ -> ´´/__#
```

Figure 2.

Then the negation of ´perfect´ is formed by first determining the first applicable prefixal substitution instruction:

- (1) is not applicable, since ´perfect´ does not begin with ´l´.
- (2) is not applicable, since ´perfect´ does not begin with ´r´.
- (3) is applicable, since ´perfect´ begins with ´p´.

The first applicable suffixal substitution instruction is the only suffixal instruction at hand, namely (5): ´perfect´ ends with ´´. By the concurrent application of (3) and (5) to ´perfect´ the target ´imperfect´ is generated, which is the negation of ´perfect´.

3. Anatomy of the System for the Automated Learning of Morphological Rules

The system is written in the programming language PL/1. It has the name PRISM, which is an acronym for ´PRogram for the Inference and Simulation of Morphological rules´.

PRISM has the macro structure shown in figure 3. At an activation of PRISM, its main procedure MONITOR at first activates GETOPTN which reads the user´s options for the control of PRISM and checks them for syntactic well-formedness and for plausibility. Then MONITOR activates the component indicated by the user´s control options. There are three alternative components:

- A learning component which infers sets of instructions from a learning corpus given by the user of PRISM. This component comprises the procedures CHKCRPS, DISCOV, STMTOUT, TODSET, and others. The learning process is performed by DISCOV. The other procedures perform peripheral functions.
- A component for the application of instructions which were inferred by the learning component. This component comprises the procedures FRODSET, APPLY, DERIVE, and others.
- A third, marginal component which prepares instructions for their printout. It consists of FRODSET, STMTOUT, and other procedures.

The activation of the learning algorithm starts with a call of CHKCRPS by MONITOR. CHKCRPS checks a given learning corpus for formal errors. The procedure activated next is DISCOV, which performs the learning processes. DISCOV first determines the different types of substitution patterns in the given learning corpus. Types of

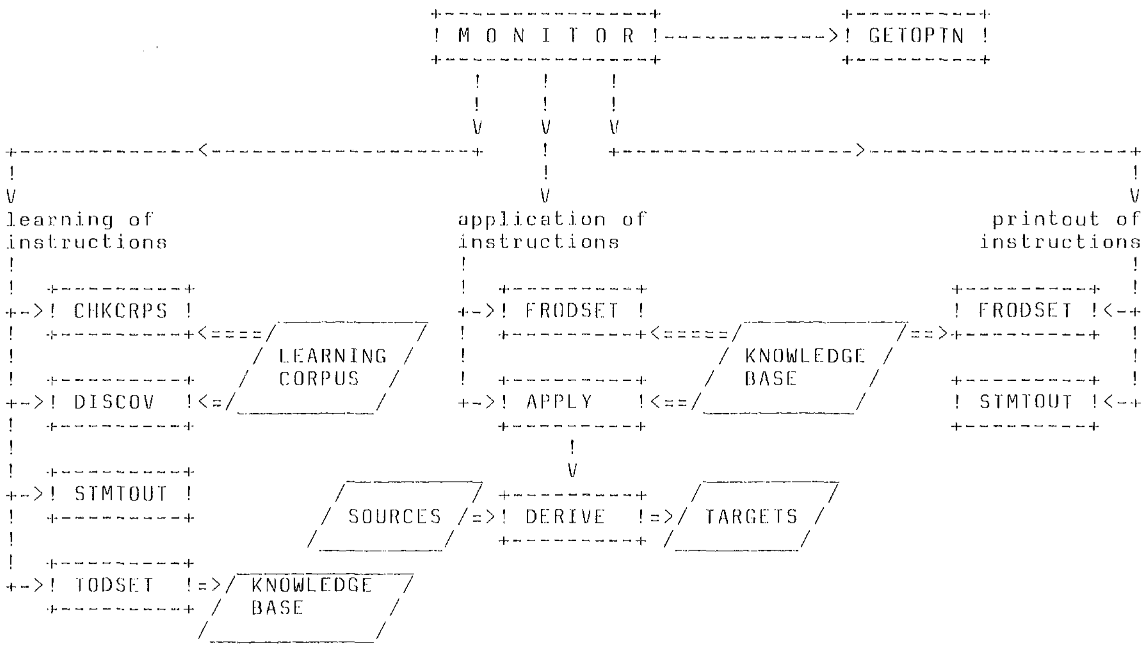

Figure 3. Macro structure of PRISM. (For reasons of lucidity some macro features of PRISM have been ignored in this chart.)

substitution patterns are the different (X, Y)-pairs which are implicitly present in the learning corpus. (For the status of X and Y compare the definition of the formalism for the representation of morphological rules.) The second step of DISCOV computes the frequency of each substitution pattern in the corpus. DISCOV's learning strategy presupposes that the substitution patterns occurring more frequently in a language also occur more frequently in the learning corpus. Therefore DISCOV creates more general instructions for the more frequent patterns of a learning corpus and more specific instructions for the less frequent patterns of a learning corpus, i. e. the contextual strings Z(i) of an instruction X -> Y/#__(Z(1)| ... |Z(i)| ... |Z(n)) or X -> Y/(Z(1)| ... |Z(i)| ... |Z(n))__# are the more general the more frequently the substitution pattern (X, Y) occurs. They are the more specific the more rarely the substitution pattern occurs. Provided that a learning corpus is representative of the morphological substitution patterns of a language and the contextual strings Z(i), this general strategy for the determination of the Z(i)'s increases the probability that the inferred instructions generate correct targets for such sources as are not elements of the given learning corpus. DISCOV arranges the substitution instructions in such a way that the more specific instructions precede the more general ones. This order of the instructions guarantees during their later application that potentially each instruction can be applied. STMTOUT transforms substitution instructions inferred by DISCOV from their internal representation, which allows their easy and fast automatic treatment, into an external representation and prints them out. For this external representation the notation is used which was introduced above in the definitions of the two types of substitution instructions. Finally TODSET stores the instructions in an external knowledge base, from which they can later be read by the other two components of PRISM (In the knowledge base the instructions are stored in their internal representation).

The application component starts with FRODSET, which loads a set of instructions to be applied from the knowledge base to the central memory. Then the two procedures APPLY and DERIVE apply the instructions to words given by the user and thereby generate targets which are written to an output data set. The kind of morphological relation between the generated targets and the given words depends on the specific set of instructions which is applied.

4. Evaluation of the System

The performance of PRISM was evaluated under the following conditions:

1. A set of instructions should always generate correct targets if it is applied to the sources of the learning corpus from which it was inferred.
2. The larger the learning corpus is for a given morphological relation, the higher should be on average the percentage of correctly generated targets for such sources as are not elements of the learning corpus (but nevertheless

participate in the given morphological relation).

3. A set of instructions inferred from a linguistically representative learning corpus should generate correct targets for at least 90% of the sources which are not elements of the learning corpus (but which nevertheless participate in the morphological relationship under discussion).
4. If a linguistically representative learning corpus is given, the learning algorithm should classify as regular those morphological patterns which linguists also usually classify as regular.

Condition 1 is fulfilled. This could be proved deductively with reference to the structure of the learning algorithm. (The proof is given in Wothke 1985, 144-154.)

The fulfilment of conditions 2-4 could only be tested inductively by applying PRISM´s learning algorithm to different learning corpora and evaluating the results.

Condition 2 was tested by applying the learning component to learning corpora of different sizes compiled for two morphological relations: derivation of nomina actionis from verbs in German (e. g.: ´betreuen´ -> ´Betreuung´), derivation of female nouns from male nouns in French (e. g.: ´spectateur´ -> ´spectatrice´). With the sets of instructions inferred from these learning corpora PRISM´s application component generated targets for a set of words not in the learning corpora. The statistical results of these tests showed that the percentage of correctly generated targets for such sources as are not elements of the learning corpus is, on average, the higher the larger the learning corpus is. A further important result was that the percentage of correctly generated targets is the higher the more regular the morphological relation is: The tests yielded better results for the more regular derivation of female nouns from male nouns in French than for the less regular derivation of nomina actionis form verbs in German.

To test the fulfilment of the third condition representative learning corpora were manually compiled for the derivation of nomina actionis from verbs in German (9.167 source-target-pairs) and for the derivation of female nouns from male nouns in French (89 source-target-pairs). The two sets of instructions automatically inferred from these two corpora were applied to large sets of sources which were not members of the learning corpora (4.793 sources for German, 211 sources for French). In both cases the percentage of correctly generated targets was 100%.

Condition 4 was tested with learning corpora for the pluralization of English nouns and for the derivation of female nouns from male nouns in French. An exact quantification of the degree of accuracy is not possible, since this condition contains some vague expressions such as "regular" and "usually". My subjective judgement is that the instructions constructed by the learning algorithm for (approximately) representative corpora are quite similar to the morphological regularities described in traditional grammars. This may be illustrated by an example: The learning corpus shown in figure 1 is approximately representative for the regular pluralization patterns of English nouns. From this corpus PRISM inferred the following set of instructions which represent the most important pluralization rules:

```
(1) ´´    -> ´´/#__

(2) ´f´   -> ´ves´/__#
(3) ´fe´  -> ´ves´/__#
(4) ´y´   -> ´ies´/(´d´|´l´|´p´|´r´|´t´)__#
(5) ´´    -> ´es´/(´ch´|´sh´|´s´|´x´|´z´)__#
(6) ´´    -> ´s´/__#
```

Figure 4.

5. Unsolved Problems

- The formalism which PRISM uses for the representation of the instructions is designed for the description of graphemic changes at the beginning and/or at the end of a word. Thus this formalism is inadequate for the description of changes in the interior of a word. These, however, occur more rarely than the changes at the beginning or at the end. A solution to this problem, which could consist in the design of a new formalism whose expressions could also be learned automatically, has not as yet been found.
- PRISM cannot recognize exceptions in a learning corpus and treat them adequately. If, for instance, the learning corpus in figure 1 would also contain the pair (´goose´, ´geese´), PRISM would infer the prefixal substitution instruction ´goo´ -> ´gee´/#__ and insert it in the set of instructions shown in figure 4 before instruction (1). Furthermore PRISM would infer the suffixal instruction ´´ -> ´´/´ose´__# and insert it before instruction (3). If this new set of instructions is applied to the nouns ´good´, ´goodness´ and ´goon´ the incorrect plurals ´geeds´, ´geednesses´ and ´geens´ are generated. - It would be preferable for PRISM to identify exceptions as such and store them in a list of exceptions instead of inferring overgeneralizing instructions from them.
- If a set of instructions is linguistically inadequate, the user of PRISM must first make the learning corpus more representative by adding suitable examples. Then he must activate the learning component of PRISM which infers a totally new set of instructions. - Perhaps it would be better if PRISM could infer new instructions only from the new examples and then synthesize these new instruc-

tions with the formerly inferred and linguistically inadequate instructions to give a new, more adequate set of instructions.

References

Cohen, P. R./Feigenbaum, E. A. (Eds.) (1982): The handbook of artificial intelligence. Vol. 3. London.

Jansen-Winkeln, R. M. (1985): Induktives Lernen von Grammatikregeln aus ausgewählten Beispielen. In: Savory, S. E. (Ed.) (1985): Künstliche Intelligenz und Expertensysteme. Ein Forschungsbericht der Nixdorf AG. 2nd ed. München/Wien. PP. 211-223.

Oakey, S./Cawthorn, R. C. (1981): Inductive learning of pronunciation rules by hypothesis testing and correction. In: Proceedings of the 7th International Jpint Conference on Artificial Intelligence. August 1981. Vol. 1. PP. 109-114.

Pinker, S. (1979): Formal models of language learning. In: Cognition 3. PP. 217-283.

Ring, H. (1978): PELIKAN - ein Lernsystem für linguistische Klassifikationsalgorithmen. In: Nachrichten für Dokumentation 6. PP. 224-226.

Wolf, E. (1977): Vom Buchstaben zum Laut. Maschinelle Erzeugung und Erprobung von Umsetzautomaten am Beispiel Schriftenglisch - Phonologisches Englisch. Braunschweig.

Wothke, K. (1984): PRISM User´s Guide. Bonn. (= IKP-Arbeitsbericht No. 5)

Wothke, K. (1985): Maschinelle Erlernung und Simulation morphologischer Ableitungsregeln. Bonn. (Doctoral dissertation).

A detailed treatment of the theme dealt with in this paper is given in Wothke (1985).