Computer Sciences Department
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

AN INTERACTIVE PROGRAM FOR LEARNING THE MORPHOLOGY OF NATURAL LANGUAGES

by

Sheldon Klein & Terry A. Dennison

Technical Report #144

December 1971

AN INTERACTIVE PROGRAM FOR LEARNING THE MORPHOLOGY OF NATURAL LANGUAGES*

Sheldon Klein & Terry A. Dennison

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706
U.S.A.

Introduction

The morphology learning program is a subcomponent of AUTOLING, a program that learns transformational grammars of artificial and natural languages through interaction with a human informant (3,4). The AUTOLING system as a whole was operational on a Burroughs B5500 computer located at the University of Wisconsin and was written in extended ALGOL for that machine. This computer is no longer available to the researchers. The AUTOLING work has been transferred to a Burroughs B6700 computer located at the University of California, San Diego, where the phrase structure learning component and portions of the morphology learning component are operational and rewritten in extended ALGOL for that machine. The transformation learning component at this date is not yet fully operational on the new machine.

Serious precursors of this work include that of Alicia Towster (4) and Paul Garvin (1). Towster's work was connected with the AUTOLING research group, but the state of development of the program did not involve a component that was integratable with the overall AUTOLING system.

---

*Research sponsored by National Science Foundation Grant GS-2595. Presented at 1971 International Conference on Computational Linguistics, Debrecen Hungary, September 4-7.

Garvin and his assistants developed an elaborate but unimplemented program design involving specific tests for semantic and morphological structure types. The methodology anticipated the usage of a knowledge of linguistic universals.

Our particular approach requires the integration of the morphology learning process with the grammar discovery methods of the total AUTOLING learning system. Primary emphasis is placed on the assumption that full automation of discovery methods will require a complete functional analysis of the semantics of the meta-language used in the learning process (in this case English). Because such an analysis is not available, glosses for forms are rewritten in a semantic notation adequate for solution to the particular problems presented. Ultimately, a program might be created that would reinterpret English in a proper semantic form automatically. Yet more would be required, for the ever present problems of meaning unit mappings between languages (several units in one language mapping into one unit of the other and vice versa, or worse) suggest a future methodology involving ultimate rewriting of glosses perhaps as universal semantic features, and a program logic capable of determining the distinctive ones for a particular language.

In the following sections we present a description of what is currently implemented or readily implementable within the framework of available computational and theoretical resources.

All discovery methods used are heuristic rather than algorithmic, which means that they are part of what has been called a linguist's "bag-of tricks" -- methods that may work, but that do not guarantee resolution of problems.

The AUTOLING program creates grammars with unordered context-free phrase structure rules coupled with ordered transformations to handle context sensitive phenomena. In the original version (without a morphology learning component) the informant was assumed to be bilingual and was required to input sentences with spaces between morphemes.

The new version does not require such preanalysis, and will yield grammars in which a transformational model is used to the level of morpheme strings. From that point on a structuralist description will be derived as well as a transformational one.

To some relationalist philosophical grammarians this may seem a strange mixture. As a logical-positivist interested in creating a system that works, the first author reserves the right to be eclectic; we note that each model is particularly suited to automation of certain discovery procedures.

The discovery heuristics of the phrase structure learning program are described in (3). We note that they involve extensive use of distributional criteria (especially frame tests). The transformation learning component makes use of informant corrections to faulty productions of the program in testing mode. Generality of specific transformations is obtained by heuristics analogous to the ones that are used in the phrase structure learning component.

The Analytic Philosophy

At every stage of the analysis, the program attempts to formulate a grammar to account for the observed data base. Accordingly, the learning heuristics must assume that the grammar is never complete, and constantly subject to revision.

This problem has been solved for the phrase structure and transformation learning components through mechanisms for creating, destroying and substituting classes of morphemes and higher level units in already existing rules.

The same capabilities applied to the morphology learning component create a separate set of data maintenance problems. We note that every input from an informant is stored and used for reference at various stages of analysis. Accordingly in an intermediate state of analysis, a new morphological breakdown of a unit previously treated as monomorphemic requires updating not only in the hierarchies of possible rule chains that may reference it, but in all stored inputs that may contain it.

A Sample Problem

The quickest explication of the methodology can be provided by an example analysis. We note that the following is a hand simulation, as the currently working portion of the system merely isolates morphs, and does not provide complex updating of existing rules. In the example many features of the system, including checks of informant glosses and this consistency are not indicated; these will be discussed in a later section.

We note that the system operates in two modes, syntactic and morphological. In the morphological mode, almost no phrase structure learning heuristics and none of the transformation learning ones are used. At the point of return to the syntax mode, all inputs that were entered during the morphological mode are reentered as inputs to the system in their analyzed, morphologically portioned form.

A major portion of heuristic strategy can govern the time the system stays in each mode, and the circumstances that demand a switch. For the following example we will adopt the rule that any time an input contains morphological material not recognized by the system, it will automatically switch to morphological mode, and return to the syntactic mode only when tentative analysis of that material has been made. We note that this is not likely to be the optimum analytic strategy.

The basic analytic method consists of double matches of forms and glosses. The matching of glosses involves simple set intersection rather than the ordering of elements that is observed when comparing forms in the language under analysis. An exception to this is possible if brackets are placed around bundles of elements in the gloss. In this case partial orderings are possible. None of the current heuristics handle problems with discontinuous morphemes. A major heuristic involves the choice of what to compare with what. Forcing the system to wait for maximally similar forms before undertaking matchings would simplify the computations.

The problem as formulated by NIDA (2) is as follows:

Problem 5 (data from the Elisabethville dialect of Congo Swahili, a language of the Belgian Congo)

Instructions:

a. List all morphemes.

b. Give the meaning of each.

1. ninasema 'I speak'
2. wunasema 'you (sg.) speak'
3. anasema 'he speaks'
4. ninaona 'I see'
5. ninamupika 'I hit him'
6. tunasema 'we speak'
7. munasema 'you (pl.) speak'
8. wanasema 'they speak'
9. ninapika 'I hit'
10. ninamupika 'I hit you (pl.)'
11. ninakupika 'I hit you (sg.)'
12. ninawapika 'I hit them'
13. ananipika 'he hits me'
14. ananupika 'he hits you (pl.)'
15. nilipika 'I have hit'
16. nilimupika 'I have hit him'
17. nitakanupika 'I will hit you (pl.)'
18. nitakapikiwa 'I will be hit'
19. wutakapikiwa 'you (sg.) will be hit'
20. ninapikiwa 'I am hit'
21. nilipikiwa 'I have been hit'
22. nilipikaka 'I hit (remote time)'
23. wunapikizwa 'you (sg.) cause being hit'
24. wunanipikizwa 'you (sg.) cause me to be hit'
25. wutakanipikizwa 'you (sg.) will cause me to be hit'
26. sitanupika 'I do not hit you (pl.)'
27. hatanupika 'he does not hit you (pl.)'
28. hatutanupika 'we do not hit you (pl.)'
29. hawatatupika 'they do not hit us'

Supplementary information:

1. The future -taka- and the negative -ta- are not related.
2. The final -a may be treated as a morpheme. Its meaning is not indicated in this series.
3. The passive morpheme may be described as having two forms -iw- and -w-. Its form depends on what precedes it (see Principles 2 and 3).

The first input is:

1. ninasema (1st person sg.) (speak present)

The system automatically jumps to morphological mode and coins the phrase structure rule:

-S1:= ninasema

A note on the form of the rules: the left hand side of each rule is given the label 'S' plus a number. A prefix of '*' indicates that the construction occurred as a free form input during the syntactic mode, a '-' prefix indicates a morpheme or morpheme class. The absence of a prefix indicates that the construction is an analytically derived higher level intermediate node.

The next input is:

2. wunasema (2nd person sg.) (speak present)

The system now tries a left alignment searching for a right match of the forms in the language under analysis (only if there is a common intersection in their glosses). If both alignments yield a match, the longest is taken. If no matches are found, the system continues to the next input.

In this case both alignments are identical:

wunasema
ninasema

The following are entered in the dictionary.

wu (2nd person sg.)
ni (1st person sg.)
nasema (speak present)

and the phrase structure rules are rewritten:

```
 S1:= S3 S5
 S2:= S4 S5
-S3:= wu
-S4:= ni
 S5:= nasema
```

At this point the system returns to the syntactic mode, and the forms are automatically reentered as 'ni nasema' and 'wu nasema'. Among other things that happen are the application of some combinatory phrase structure rules that operate on rules S1 and S2 which have now been given asterisked, free form status. The resultant grammar is:

```
*S1:= S6 S5
-S3:= wu
-S4:= ni
-S5:= nasema
 S6:= S3
 S6:= S4
```

The next input is:

3. anasema (3rd person sg.) (speak present)

Again the system switches to morphological mode. At this point, the dictionary is used to parse the input, i.e. to identify previously determined morphemes on the basis of longest embedded matches, and set inclusion of the dictionary gloss in the total input gloss. 'nasema' is found, and 'a (3rd person sg.)' is added to the dictionary. The following rules are subsequently added to the phrase structure grammar:

```
 S7:= S8 S5
-S8:= a
```

An attempt is made to find this new 'a' in other dictionary entries, but it fails because of a lack of semantic intersection.

After a return to syntax mode, rule S7 is deleted and

S6:= S8

is added.

The next input is:

4. ninaona (1st person sg.) (see present)

The dictionary check yields 'ni (1st person sg.)' and 'naona (see present)' is added to it.

At this point, the newly entered item is matched against other dictionary entries, yielding the common element:

na present

and the newly segmented disjunctive items:

sema speak
ona see

The final result after reentry into syntax mode includes a rewriting of rule S5 and a combination of 'ona' and 'sema' into one class:

```
*S1:= S6 S5
-S3:= wu
-S4:= ni
 S5:= S9 S10
-S6:= S3
-S6:= S4
-S6:= S8
-S8:= a
-S9:= na
-S10:= sema
-S10:= ona
```

The elements of the next input:

5. ninamupike (1st person sg.) (hit present) (3rd person sg.)

are all accounted for by the dictionary except for 'mu' which is entered with the gloss '(3rd person sg.)'. At this point we might have written the gloss for 'mu' in the original input as '(3rd person sg. object)'. Having done the problem in advance, we know that 'mu' and 'a' should eventually be combined as allomorphs, making the specification of 'object' in the semantics superfluous.

This suggests that the system must have heuristics capable of determining non-distinctive semantic features if there is over-specification.

The anticipated tactic at this point is to avoid attempts at immediate resolution, and to treat the two morphemes as independent entities. The resultant grammar offers further processing in both morphological and syntactic modes yields the addition of two rules:

```
*S11:= S6 S9 S12 S10
-S12:= mu
```

However the syntax mode heuristics continue to analyze the rules and seek to combine the partially similar S1 and S11, resulting in the deletion of S11, and the rewriting of S1 as

```
*S1:= S6 S13
```

and the addition of two new rules:

```
S13:= S5
S13:= S9 S12 S10
```

The input form:

6. tunasema (1st. person plu.) (speak present)

adds segment 'tu' to the dictionary, and the rule:

-S14:= tu

Input 7: munasema (2nd person plu.) (speak present)

at first yields mu (2nd person plu.) as a dictionary entry. Comparison with the already existing entry 'wu (2nd person sg.)' yields a re-analysis, resulting in the new dictionary entries:

u (2nd person)
w sg
m pl

It should be possible for the reader to anticipate the future course of the analysis. The program will split some pronoun morphemes into two components and leave others as single units. 'li past', 'taka future' and 'pika see' will be cut, and the system will decide that 'pik see' is also a morph when determining 'iwa passive' and 'wa passive'. 'ka remote time' will be cut and 'si', 'ha', 'hatu' and 'hawa' will all attain pronoun morph status at the time 'ta negative' is cut. The reader, of course, might wish to handle the problem somewhat differently, but the program cannot take Nida's hint about the final 'a' because no meaning is indicated, and the negative is unsatisfactorily solved partly for this reason and and partly because of the program's inability to segment into discontinuous morphemes.

Hierarchies of Heuristics

The hand simulation we have been doing does not indicate all the testing done by the system. Especially, it does not indicate the yet to be programmed heuristics that will monitor and govern the basic segmentation program described above.

As in the fully functioning syntax learning component, a basic design principle is the use of higher level analytic components to analyze and perhaps reject tentative results of lower level, brute force heuristics. Some of the blocking criteria can be very specific and even stylistic. In the preceeding problem one might wish to block the segmentation of the pronoun system into person and number for special reasons. The pronoun systems of many languages lend themselves to very complicated segmentation of little or no generality; many linguists prefer to avoid such cutting and prefer to treat as single units entities that might otherwise be segmented.

Another key heuristic involves the avoidance of comparison of forms for segmentation pruposes except under conditions likely to produce optimum results: such heuristics might require maximal similarities in shape and glosses within a minimum size sample.

Perhaps the most powerful heuristics used have not yet been indicated; they involve the testing of the grammar at each stage of rule modification through the generation of test productions whose generative history includes the newly created or modified rules. Such testing permits phrase structure rule modification, or may lead to the learning of a transformation if the program should require the informant to supply a correction.

Testing in the system under construction includes a translation of the test production that is also offered to the informant for acceptance or rejection. Accordingly, the informant has five possible responses: acceptance of test production and translation, rejection of both with refusal of correction, rejection of both with correction of both, correction just for gloss, and correction of just the form, with acceptance of the gloss.

The kind of corrections provided by the informant provides the data for determining allomorphic status and the coining of morphophonemic rules. Of course it is possible to develop the program in such a way that the morphology is handled implicitly in the form of transformations. However accidents in the input sequence predictably can lead to situations where only the analytic techniques of structuralist taxonomic morphological analysis will be able to recover the pertinent data. Once such techniques have been applied, it is possible to reformulate the information in a transformational model.

The heuristics for discovering morphophonemic relations can involve the use of phonological distinctive features, set intersection and resultant generalization.

The use of an articulatory phonetic chart, in the form of an array in several dimensions,also lends itself to use in a powerful heuristic for the extension of generality of morphophonemic rules.

Given an established morphophonemic rule involving a single phonemic unit, similar rules that involve members of the same row or column in the articulatory array may be tested. A similar heuristic could be obtained from distinctive feature usage, but array row computation might be faster than set intersection of distinctive features. Also, the heuristic in array form might yield hypotheses of greater

phonological plausibility at an earlier stage.

For taxonomic unification of morphs into morphemes, the system provides outstanding information about distribution. Two morphs with identical glosses but variant shapes initially will have different phrase structure rule numbers assigned to them. The existing program maintains an inverse index of all rules containing a given class descriptor. Accordingly retrieval of all constructions involving a particular morph is simple. Environments of candidates for merger into a single morpheme are readily compared.

We have a reluctance to abandon any heuristics that may be peculiar to a specific grammatical model just for the sake of theoretical purity. Actually, it seems worthwhile to permit the system to formulate morphological treatment in both transformational and taxonomic models, and to provide mechanisms for convertibility just to preserve the heuristic techniques available to each formulation.

A Note on Semantics and Meta-languages

The construction of a high quality morphological analysis program will undoubtably prove to be more difficult than the task of automating all other aspects of grammar discovery. Given even complete control over the semantics of the language in which the glosses are formulated, given even a well developed theory of universal semantic features, the problems of learning the mappings of meaning units from one language to another in the general case is quite difficult. (Let us define the general case as consisting of a language situation wherein an utterance of m morphemes containing n semantic units is translated by a gloss containing p morphemes representing q semantic units -- and where m, n, p, q may take

on any independent integer values). Of the work actually done in this area, the program designs of Garvin are the most developed, although unimplemented and untested (1).

A fruitful approach, in conjunction with other methods, would well include attempts to perform simultaneous analyses of both a language and the semantic structure of related glosses.

The Whorfian hypothesis, the notion that the structure of a language determines the speaker's perception of the universe, is at least partially antithetical to a notion of semantic universals. The implication of the Whorfian hypothesis for discovery methodology is an inversion of the original formulation: a knowledge of the extralinguistic universe of a language speaker is a prerequisite to a knowledge of the semantic structure of his language. In absence of proof or disproof of the total accuracy of either view, an empirical researcher must be prepared to analyze linguistic situations involving a little of both. Indeed, even if the universalist position is the correct one, the techniques of an approach assuming non-universality cannot yield false results, but rather corrorborative data.

There is one area in linguistic analysis where the first author will vigorously defend the validity of the Whorfian hypothesis -- the meta-languages associated with grammatical descriptions. Few linguists (if any, acknowledge that at least two meta-languages are associated with every linguistic description. Many might acknowledge the language of gloss representation as one, but few would concede that the theoretical model used to formulate the description is really another. In each of the two, the structure determines the linguist's perception of the realities

of the language it is used to describe.

One may bring yet a third meta-language to the scene with a tale of the first author's experience that the choice of programming language in computational linguistic work can alter radically the structure of the solutions to particular problems. Often the choice of a particular programming language can make theoretical problems that appear difficult in their original formulation seem trivial in their programmed treatment and vice versa.

At this point the reader can guess the implied methodology: with regard to theories, linguists should be exploitive masters rather than servants.

Bibliography

1. Garvin, Paul L. Computer-Based Research on Linguistic Universals. Bunker-Ramo Corporation Quarterly Progress Reports, under contract NSF 576, Series G096-8U3, Canoga Park, California, 1967-70.

2. Nida, E. A. Morphology, the Descriptive Analysis of Words, 2nd Edition, University of Michigan Press, Ann Arbor, 1949.

3. Klein, S. & Kuppin M. A. "An Interactive Heuristic Program for Learning Transformational Grammars." University of Wisconsin Computer Science Department Technical Report #97, August 1970. Also in press, Journal of Computer Studies in the Humanities & Verbal Behavior.

4. Klein, S., Fabens, W., Herriot, R. G., Katke, W. J., Kuppin, M. A., & Towster, A. E., "The AUTOLING System", University of Wisconsin Computer Sciences Department Technical Report #43, September 1968.