VLADIMIR PERICLIEV

THE PROSPECTS FOR MACHINE DISCOVERY IN LINGUISTICS

ABSTRACT. The article reports the results from the development of four data-driven discovery systems, operating in linguistics. The first mimics the induction methods of John Stuart Mill, the second performs componential analysis of kinship vocabularies, the third is a general multi-class discrimination program, and the fourth finds logical patterns in data. These systems are briefly described and some arguments are offered in favour of machine linguistic discovery. The arguments refer to the strength of machines in computationally complex tasks, the guaranteed consistency of machine results, the portability of machine methods to new tasks and domains, and the potential machines provide for our gaining new insights.

KEY WORDS: linguistic machine discovery, scientific discovery, prospects for computational discovery

1. INTRODUCTION

Scientific discovery was one of the favourite topics of Renaissance scholars like F. Bacon, Descartes, and Leibnitz. These early efforts suffered a long period of oblivion (basically, due to the critiques of Hume and Whewell), but this century has witnessed a steady revival of interest. In his classic *The Logic of Scientific Discovery*, Popper indeed denounced a logic of discovery (in contrast to justification) but works like that of Hanson (1958) offered convincing counterarguments, rehabilitating discovery in the eyes of the scientific community as a whole. Significant contributions to a general understanding of the discovery process have come from diverse scientific disciplines, like mathematics (Hadamard, Polya), psychology (Wertheimer, Duncker), and philosophy (Nickles, 1980a, b). Cognitive science and AI were also very instrumental in this regard, providing computer models of rational reconstruction or tools enhancing novel discoveries (cf. esp. Newell

Foundations of Science **4:** 463–482, 1999.

and Simon (1972); Langley et. al. (1987); Shrager and Langley (1990); the *Artificial Intelligence* journal Special Issue on Machine Discovery of April 1997). In effect, in many disciplines to date, discovery is considered quite a respectable object of investigation.

The topic of discovery, however, is not a popular one in linguistics to date. There are a number of reasons for that, but, looking retrospectively, two seem to be especially important (for a more detailed discussion, cf. Pericliev (1995)). The first reason is historical and is connected with the severe (and often unjust) critiques on the part of the emerging, but quickly becoming influential, transformational school. Thus, more than 20 years after the original publication of Popper's *The Logic of Scientific Discovery*, Noam Chomsky (1957, p. 56) stated in his *Syntactic Structures*, in words closely reminiscent of Popper, that the primary goal of linguistics is justification, not discovery, and that "... it is questionable whether [discovery procedures] can be formulated rigorously, exhaustively and simply enough to qualify as practical and mechanical discovery procedure". These claims had the effect in linguistics that Popper had earlier produced in philosophy, but, unfortunately, unlike philosophy (cf. esp. Nickles (1980a, b) and subsequent work) linguistics never managed to recover from this blow.

The other reason is a naïve sort of empiricism, widely practised in linguistics, recognizing, correctly, the need for gathering large bulks of diverse data about human languages, and investing much effort in this enterprise, but still failing to conceive the need for working out efficient methods for mining the wealth of knowledge already acquired.

In this article, I will argue for the prospects of machine discovery in linguistics from the position of both a practising linguist and one working on machine discovery (for a terse general comment, cf. also Pericliev (1996); Pericliev (1990) illustrates the use of some heuristics on two research problems). I shall focus on discovery from data, basing my discussion on the results from four discovery systems in linguistics I have been involved in during the last few years. Section 2 is an outline of these systems, and Section 3 offers some arguments in favour of machine discovery in linguistics, pertaining to the strength of machines in computationally complex tasks, the guaranteed consistency of machine results, the portability of machine methods to new tasks and domains, and the potential machines provide for our gaining new insights into the tasks modelled. Finally, the conclusion is drawn that more efforts need to be invested in (mechanical) linguistic discovery.

2. FOUR DISCOVERY SYSTEMS

This section is a brief overview of four linguistic discovery systems, conducted basically in terms of their input/output, and some of the discoveries they have made.

2.1. *The system MILL*

MILL (Pericliev, 1995) is a system designed to mimic the induction methods (“canons”) of John Stuart Mill (Mill, 1879) for discovery of “causally” related facts in a set of instances (observations). Mill’s methods assume that each observation consists of a set of putative causes (“accompanying facts” or “circumstances”) for an effect; their aim, as eliminative induction methods, is to eliminate, from the set of putative causes, all but the “actual” one.

Below we state three of Mill’s heuristics (leaving out of consideration the fourth, the Method of Concomitant Variation), following his own formulation:

The Method of Agreement: If two or more instances of the phenomenon under investigation have only one circumstance in common, the circumstance in which alone all the instances agree, is the cause of the given phenomenon. Schematically:

$$\begin{array}{c} A, B, C \rightarrow a, b, c \\ A, D, E \rightarrow a, d, e \\ \hline A \text{➜} a \end{array}$$

(Here, and in the following, “→” means “accompanies”, “➜” means “causes”, capital letters denote circumstances (causes), and small-case letters the “phenomena” (effects) investigated.)

The Method of Difference: If an instance in which the phenomenon under investigation occurs, and an instance in which it does not occur, have every circumstance in common save one, that one occurring only in the former, then the circumstance in which alone the two instances differ is the cause, or an indispensable part of the cause, of the phenomenon. Schematically:

$$\begin{array}{c} A, B, C \rightarrow a, b, c \\ B, C \rightarrow b, c \\ \hline A \boldsymbol{\rightarrow} a \end{array}$$

The Method of Residues: Subduct from any phenomenon such part as is known by previous inductions to be the effect of certain antecedents, and the residue of the phenomenon is the effect of the remaining antecedents. Symbolically:

$$\begin{array}{c} A, B \rightarrow a, b \\ B \rightarrow b \\ \hline A \boldsymbol{\rightarrow} a \end{array}$$

Considering a somewhat simplistic linguistic example, assume that we are given the morpheme decomposition of (English) words, as well as their decomposition into constituent meanings, and that we inquire about the morpheme-meaning correspondences. By the Method of Agreement we may infer from the following observations that the morpheme “let” and the meaning “diminutive” are causally connected:

$$\begin{array}{c} \text{book,let} \rightarrow \text{‘book’, diminutive} \\ \text{leaf,let} \rightarrow \text{‘leaf’, diminutive} \\ \text{book,let,s} \rightarrow \text{‘book’,diminutive} \\ \hline \text{let} \boldsymbol{\rightarrow} \text{diminutive} \end{array}$$

Thus, since only the morpheme “let” occurs when the resultant meaning (= effect) “diminutive” occurs, while the other accompanying facts (“book”, “leaf” or “s”) vary, we conclude that “let” causes the meaning “diminutive”.

The system MILL incorporates these methods. Basically, its discovery process is an attempt to apply one or more of the methods, iterating through the database. After each successful application of a method, the system tries to confirm the conjecture by testing its validity in the entire database. A successful conjecture is then attempted to be proven by a further method (which will further

TABLE I

Indo-European and Germanic cognate words (input)

No.	Causes	Effects	
No.	Causes	Effects	
1.	t,u	ð,ū	(you)
2.	t,r,ē,s	ð,r,e,e	(three)
3.	p,a,t,e,r	f,a,ð,e,r	(father)
4.	p,ē,s	f,ō,t	(foot)
5.	p,e,c,u	f,e,o,h	(fee,cattle)
6.	n,e,p,ō,s	n,e,f,a	(nephew)
7.	d,u,o	t,w,a	(two)
8.	d,e,c,e,m	t,ē,n	(ten)

increase its plausibility) and the result is recorded. The system repeats this process until all possibilities are exhausted.

As an illustration of the system, I show its re-discovery of (a part of) the famous phonological law, known as Grimm’s Law (1822), pertaining to a consonantal shift, taking place in the Germanic languages, from the original Indo-European consonants. The latter are taken as the causes of the later Germanic developments. In Table I, showing the input to the system, the Indo-European sounds are exemplified by Latin (left hand side) and the Germanic sounds by Old English words (right hand side; the translations in the last column serve for clarity, and are not part of the input).

Running the system on the above database, it has found the consonantal alternations: $t > ð, p > f, d > t$, which constitute a part of Grimm’s Law. Besides these causally related pairs, MILL will provide an elementary form of “explanation”, stating the methods and data used to draw an inference. In this particular case it will inform the user, say, that the conclusion about the shift $t > ð$ was arrived at, applying the Method of Agreement to the observations 1, 2, and 3, and that no further method can prove this conclusion.

MILL has re-discovered a number of further sound laws, and has also been used as an aid in the solution of diverse linguistic field problems (finding translation equivalents in aligned texts, etc.)

2.2. *The system KINSHIP*

The KINSHIP program (Pericliev and Valdés-Pérez, 1998a, c) models the componential analysis of kinship vocabularies. This topic was a very popular one in both linguistics and social anthropology in the near past, and is still of interest today. The basic reason is that kinship vocabularies of the world's languages exhibit diverse linguistic semantic structures (i.e. different partitioning of the space of relatives by the kin terms), which find reflexes in the social organization of the society speaking a language.

The system accepts as input data the kin terms with their attendant relatives, stated in a traditional notation: Fa = father, Mo = mother, Br = brother, Si = sister, So = son, Da = daughter, Hu = husband, Wi = wife. More complex kinship relations are expressed by juxtaposing these primitive relations, e.g. MoFa (mother's father), WiBr (wife's brother), etc. The output of the system is a bundle (conjunction) of feature-value descriptions of the kin terms of a language such that each kin term meaning is distinguished from all the rest by at least one feature-value, called "componential analysis". Two simplicity constraints – familiar from the linguistic/anthropological literature – are imposed on the componential analyses produced. Thus, the system uses: (1) the least number of overall features sufficient to discriminate every kin term from all the rest, and (2) the least number of features for each kin term meaning description. Besides these simplest analyses, KINSHIP can produce all other (i.e. non-simplest) alternative analyses in order to support analysts' concerns other than parsimony.

As an illustration, we may look at the Swedish consanguineal (blood) kin terms. Cf. Table II.

Running KINSHIP on these data will produce the componential analysis given in Table III. The analysis employs the following features (some of which are self-explanatory):

(1) The binary feature +/– male, showing the sex of the relative.

(2) The binary feature +/– male-1st-link, showing the sex of the first connecting relative (e.g. *morfar*, mother's father, is – male-1st-link, since the first connecting relative, viz. mother, is female, while *farfar*, father's father, is +male-1st-link, since the first link, viz. father, is +male).

TABLE II

Swedish consanguineal terms with their attendant relatives (input)

Kin terms	Relatives
fader	Fa
moder	Mo
farfar	FaFa
morfar	MoFa
farmor	FaMo
mormor	MoMo
son	So
dotter	Da
broder	Br
syster	Si
farbror	FaBr
morbror	MoBr
faster	FaSi
moster	MaSi
sonson	SoSo
dotterson	DaSo
sondotter	SoDa
dotterdotter	DaDa
brorson	BrSo
systerson	SiSo
brordotter	BrDa
systerdotter	SiDa
kusin	FaBrSo MoBrSo FaSiSo MoSiSo FaBrDa MoBrDa FaSiDa MoSiDa

(3) The multi-valued numerical feature generation.

(4) The multi-valued numerical feature genealogical distance, expressing the number of consanguineal links (primitive relations) between ego (the speaker) and the designated relative (e.g. *brorson*, brother's son, is geneal_distance = 2, since ego is removed two links from the designated relative, while *kusin*, father's brother's son, etc., is geneal_distance = 3).

TABLE III

Componential analysis of the Swedish consanguineal terms (output)

fader	+male & generation = 1 & geneal_distance = 1
moder	–male & generation = 1 & geneal_distance = 1
farfar	+male-1st-link & +male & generation = 2
morfar	–male-1st-link & +male & generation = 2
farmor	+male-1st-link & –male & generation = 2
mormor	–male-1st-link & –male & generation = 2
son	+male & generation = –1 & geneal_distance = 1
dotter	–male & generation = –1 & geneal_distance = 1
broder	+male & generation = 0 & geneal_distance = 1
syster	–male & generation = 0 & geneal_distance = 1
farbror	+male-1st-link & +male & generation = 1 & geneal_distance = 2
morbror	–male-1st-link & +male & generation = 1
faster	+male-1st-link & –male & generation = 1
moster	–male-1st-link & –male & generation = 1 & geneal_distance = 2
sonson	+male-1st-link & +male & generation = –2
dotterson	–male-1st-link & +male & generation = –2
sondotter	+male-1st-link & –male & generation = –2
dotterdotter	–male-1st-link & –male & generation = –2
brorson	+male-1st-link & +male & generation = –1 & geneal_distance = 2
systerson	–male-1st-link & +male & generation = –1
brordotter	+male-1st-link & –male & generation = –1
systerdotter	–male-1st-link & –male & generation = –1 & geneal_distance = 2
kusin	geneal_distance = 3

The above analysis reflects the rules speakers of the language should be aware of in order to use correctly the kin terms of the language. Furthermore, the meaning description of terms is the most parsimonious one in that 4 are the minimum number of features needed to accomplish the discrimination, and each kin term is expressed with the minimum number of feature-values (e.g. one feature-value, viz. geneal_distance = 3, is sufficient to demarcate *kusin* from all remaining terms).

I might mention some of the facilities of KINSHIP that assist the user of the system in the study of kinship systems. First, as regards the input data, KINSHIP maintains subroutines that check their “correctness” (no relative should be associated with more than one kin term), and the user can query the system in a variety of ways, e.g. what relatives are associated with a given kin term; given a relative, what is its corresponding kin term, etc. As regards semantic features, the user can obtain information as to what set of features are currently used by the system, and eventually select (for some reasons, e.g. psychological or social) a subset of these to accomplish the analysis. Also, one could retrieve what relative/kin term possesses what features. The user can inquire about the semantic contrasts existing between a selected pair of kin terms or check what kin terms contrast with respect to a selected contrasting feature.

There are two basic steps in the algorithm. In the first step, the feature-values of kin terms are computed, which are the values shared for some feature by all attendant relatives of a kin term, e.g. all relatives designated by *kusin*, FaBrSo, MoBrSo, FaSiSo, MoSiSo, etc., have genealogical-distance = 3. To do this, the system is endowed with a set of semantic features like the above ones which currently number about 30. The second step of the algorithm uses the kin term descriptions in term of feature-values, obtained at step 1, to find a minimum number of overall features needed to discriminate the domain, and the minimum number of feature-values for each kin term.

Some discoveries of the program are well worth mentioning. KINSHIP has so far been applied to the analysis of more than 20 languages of different language families. The program has rediscovered the meaning structure of terms of the American Indian language Seneca, as found in a classical article by Lounsbury (1964), which was a major advance in understanding the Iroquois family relationships, showing at the same time the accuracy of his analysis. The program was applied to unanalysed kinship terminologies (e.g. Bulgarian), and has improved on previous componential analyses (e.g. English). Thus, for English KINSHIP has found the most parsimonious analysis known in the literature, and the only one that manages to give conjunctive (rather than occasionally, disjunctive) definitions of all kin terms.

We have also tested componential analyses proposed by human kin analysts, revealing in some cases their logical inconsistency. E.g.

we defined in our system the same features and used the same data as Nogle (1974), addressing the English kinship system. The program produced a different model than that proposed by Nogle, showing two deficiencies in his proposal. First, he mistakenly believes that it is not possible to contrast some kin terms with the available features, whereas we found perfectly legitimate term definitions. Secondly, again mistakenly, he offers definitions of some kin terms, that cannot in fact be discriminated with the features he employs.

We also noted that, despite the widely proclaimed goal of componential analysis to generate the most parsimonious models, some practitioners actually violate simplicity constraint (2) above, and offer redundant definitions of kin terms, while others indeed stick to non-redundant (simplest) ones. Two kinds of redundancy were isolated according to whether or not the values for all overall features are included in a term's definition, i.e. a "fully redundant", and "partially redundant" definition, respectively. The system incorporates all three styles of definitions.

Another observation led to a further development of the program. There was a common belief in the field that there is only a limited number of empirically possible componential models for one kinship system, though theoretically this number may be without limit. KINSHIP showed that not only the theoretical, but also the empirical, possibilities, arising from running the program on some kin systems with the available features, may be immense in number; the two simplicity constraints used were far too insufficiently restrictive. It was observed that this undesired proliferation of models stems from a lack of coherence between the alternative definitions in a model. So two further "coherence constraints" were implemented. It will take me too far to describe them here, and I will only mention that they are quite intuitive in nature, and in fact manage to prune drastically the admissible solutions to one or just a few models.

2.3. *The MPD program*

The Maximally Parsimonious Discrimination program (MPD) is a general computational tool for inferring, given multiple classes (or, a typology), with attendant instances of these classes, the profiles (= descriptions) of these classes such that every class is contrasted from all remaining classes on the basis of feature values (cf. Valdés-Pérez and Pericliev, 1997; Pericliev and Valdés-Pérez, 1998b). In essence, MPD comprises the second, class-discrimination, module of KINSHIP, extended, however, by further possibilities.

The MPD program uses Boolean, nominal and numeric features to express contrasts between classes (two classes C1 and C2 contrast by a feature if the instances of C1 and the instances of C2 do not share a value for that feature).

MPD distinguishes two types of contrasts: (1) *absolute contrasts*, when all the classes can be cleanly distinguished, and (2) *partial contrasts*, when no absolute contrasts are possible between some pairwise classes, but absolute contrasts can nevertheless be achieved by deleting up to *N* per cent of the instances, where *N* is specified by the user.

The program can also invent *derived features* – in the case when no successful (absolute) contrasts are so far achieved – the key idea of which is to express interactions between two given primitive features. Thus:

- Two Boolean features P and Q are combined into a set of two-place functions, none of which is reducible to a one-place function or to the negation of another two-place function in the set. The resulting set consists of $P \wedge Q$, $P \vee Q$, $P \Leftrightarrow Q$, $P \Rightarrow Q$, and $Q \Rightarrow P$.
- Two nominal features M and N are combined into a single two-place nominal function $M \times N$.
- Two numeric features X and Y are combined by forming their product and their quotient.

Like KINSHIP, MPD minimizes the overall features and profiles to guarantee the uncovering of the most parsimonious discrimination among classes, and generates all alternative solutions. It also employs one of the coherence constraints of KINSHIP that seems general enough to be applied outside of the kinship domain.

By way of a simple example, illustrating most of the capabilities of the system, let us consider the discovery of the Bulgarian translational equivalents of the English verb *feed* on the basis of the syntactic frames of this verb (the uncovering of translational equivalents in the target language, based on the syntactic use of the verb in the original language, is a standard task in translation theory).

Assume the following features/values, describing the syntactic use of *feed*, pertaining to its subject (NounPhrase1), the verb itself, the object (NounPhrase2), and the Prepositional Phrase, respectively: (1) NP1 = {hum,beast,phys-obj}, (2) VTR (binary feature denoting whether the verb is transitive or not), (3) NP2 (same values as NP1), (4) PP (binary feature expressing the obligatory presence of a prepositional phrase). An illustrative input to MPD is given in Table IV (the sentences in the third column of the table are not a part of the input, and are only given for the sake of clarity though, of course, they would normally serve for deriving the instances via parsing).

The output of the program is given in Table V. MPD has successfully discriminated all classes. This is done by the overall feature set {NP1, PP, NP1 × NP2}, whose first two features are primitive and the third is a derived nominal feature. Not all classes are absolutely discriminated: Class 4 (*zaxranvam*) and Class 5 (*podavam*) are only partially contrasted by the feature NP1. Thus, Class 5 is 66.6% NP1 = phys-obj since we need to retract 1/3 of its instances (particularly, sentence (3) from Table IV whose NP1 = hum) in order to get a clean contrast by that feature. Class 1 (*otglezdam*) and Class 2 (*xranja*) use in their profiles the derived nominal feature NP1 × NP2; they actually contrast because all instances of Class 1 have the value ’hum’ for NP1 and the value ’beast’ for NP2, and hence the “derived value”’ [hum beast], whereas *neither* of the instances of Class 2 has an identical derived value (indeed, referring to Table IV, the first instance of Class 2 has NP1 × NP2 = [hum hum] and the second instance NP1 × NP2 = [beast beast]). The resultant profiling on Table V is the simplest in the sense that there are no more concise overall feature sets that discriminate the classes, and the profiles – using only features from the overall feature set – are the shortest.

MPD has been applied to a variety of tasks from different linguistic disciplines (Pericliev and Valdés-Pérez, 1998b). These include lexical semantics (discrimination of lexical meanings of sets of related words in terms of their meaning components), phonology (discrimination of the phonemes of a language in terms of distinctive features), language typology (discrimination of languages in terms of their word order), language disorders (profiling aphasic syndromes), and historical linguistics (discrimination of language families in terms of their kinship semantics).

TABLE IV

Classes and their instances (input)

Classes	Instances	Illustrations
1. otglezdam	1. NP1 = hum VTR NP2 = beast ¬PP	1. He feeds pigs
	2. NP1 = hum VTR NP2 = beast ¬PP	2. Jane feeds cattle
2. xranja	1. NP1 = hum VTR NP2 = hum ¬PP	1. Nurses feed invalids
	2. NP1 = beast VTR NP2 = beast ¬PP	2. Wild animals feed their cubs regularly
3. xranja-se	1. NP1 = beast¬VTR PP	1. Horses feed on grass
	2. NP1 = beast¬VTR PP	2. Cows feed on hay
4. zaxranvam	1. NP1 = hum VTR NP2 = phys-obj PP	1. Farmers feed corn to fowls
	2. NP1 = hum VTR NP2 = phys-obj PP	2. This family feeds meat to their dog
5. podavam	1. NP1 = phys-obj VTR NP2 = phys-obj PP	1. The production line feeds cloth in the machine
	2. NP1 = phys-obj VTR NP2 = phys-obj PP	2. The trace feeds paper to the printer
	3. NP1 = hum VTR NP2 = phys-obj PP	3.Jim feeds coal to a furnace

Among the noteworthy discoveries of MPD is that of the most economic profiling of the Russian phonemic system. In a famous article, Cherry et. al. (1953) attempted to find the most succinct description of the forty two Russian phonemes in terms of a set of universally valid binary features (e.g. consonantal, voiced, etc.). MPD confirmed that indeed eleven overall features are needed, as claimed in the article, but also found that, on the average,

TABLE V

Classes and their Profiles (output)

Classes	Profiles
1. otglezdam	¬PP & NP1 × NP2 = ([hum beast])
2. xranja	¬PP & NP1 × NP2 = ([hum hum] ∨[beast beast])
3. xranja-se	NP1 = beast PP
4. zaxranvam	NP1 = hum PP
5. podavam	66.6% NP1 = phys-obj & PP

6.2 features per phoneme profile are needed rather than 6.5, as suggested by these authors. Our program discovered shorter profiles for eleven phonemes out of a total of forty two phonemes.

2.4. *The UNIV program*

UNIV is a program for the discovery of “patterns” in data. Given a set of observations of objects, where these objects are described in terms of feature-values, the program forms and checks the validity of the logical expressions (“patterns”) in all the data:

(1) A

(2) $A \wedge B$(conjunction)

(3) $A \vee B$(disjunction)

(4) $A \Rightarrow B(A_1 \wedge A_2 \wedge \ldots \wedge A_n \Rightarrow B)$(implication)

(5) $A \Leftrightarrow B$(equivalence)

When one is searching for patterns in substantial bulks of data in which the objects are individual languages, this study is referred to as the study of “language universals” in linguistics (hence the name of the system). The search for language universals is a significant issue in contemporary linguistics, since it will reveal the properties shared by all languages and thus can help gain a better understanding of the central notion “human language”.

A simple example will suffice to get an idea about what the system does. Consider the miniature input in Table VI, describing just three languages in terms of three binary features, pertaining to the presence/absence of words, free word-order, and cases on nouns.

TABLE VI

Several languages and their descriptions (input)

Objects	Descriptions
English	+words –free_word_order –cases
Latin	+words +free_word_order +cases
Bulgarian	+words +free_word_order –cases

The system will infer e.g. that: (1) All languages have words; (2) A language has no cases or has free word order; (3) If a language has cases then it also has free word order (universals conforming to Patterns 1, 3, and 4, respectively).

UNIV has been applied to a database consisting of thirty languages, of wide genetic and areal coverage, compiled in a seminal paper by Greenberg (1966). The languages are described in terms of fifteen word order relations (mostly binary), such as adjective $<$ noun, numeral $<$ noun, demonstrative pronoun $<$ noun, use of prepositions (rather than postpositions), etc. Interestingly, the system has discovered many more universals than Greenberg himself and others (e.g. Hawkins 1983), using that database, have been able to find. Below, I give just a few of the newly discovered implicational universals holding between two variables (i.e. conforming to the Pattern A $\Rightarrow$ B). (A flood of novel implicational universals crops up, holding among three variables, i.e. of the Pattern $A_1 \wedge A_2 \Rightarrow B$.)

Universal 1: If the noun in a language precedes its modifying genitive, then this language is prepositional.

Universal 2: If the noun in a language precedes its modifying genitive, then the noun also precedes the relative clause that modifies it.

Universal 3: If a language has the order Verb-Subject-Object, then the noun precedes the relative clause that modifies it.

Universal 4: If a language uses suffixes (rather than prefixes), then the genitive precedes the noun it modifies.

Universal 5: If the demonstrative pronoun in a language follows the noun it modifies, so does the adjective.

Universal 6: If a language uses prefixes rather than suffixes, then it will have the order Subject-Verb-Object.

The idea of automated search for universals (curiously not exploited so far in linguistics) has been extended to automated searching for non-random, or statistically significant, universals in Valdés-Pérez and Pericliev (1999).

Importantly, recently the author has extended UNIV with a text generation program that expresses its discoveries in natural language. Before long we may expect discovery programs that write scientific articles about their discoveries.

3. ARGUMENTS IN FAVOUR OF MACHINE LINGUISTIC DISCOVERY

What are the lessons to be learned from these discovery systems operating in linguistics? Below I outline briefly four arguments as to why machine tools may be essential in the conduct of linguistic inquiry.

3.1. *The strength of machines in computationally complex tasks*

My first argument relates to the fact that linguistic problems not infrequently involve great bulks of data and painstaking computations. Thus, all our systems perform exhaustive searches in very large solution spaces, often dense with solutions, searches that are very difficult, or even impossible, to perform by a human analyst. UNIV for instance, run on Greenberg's database, constructs and then checks in the data about 10,000 hypotheses to find several hundred solutions. (As regards implications, the search was limited to patterns comprising only two and three variables, disregarding those with more variables, which would have significantly enlarged the solutions space.) This sample, consisting of just 30 languages, is by no means a big one; to date, there exist similar "word order" databases comprising about 500 languages described in terms of about a dozen features, which would increase the complexity of the computations much further. Even the miniature data set in Table 6 involves the formation of more than 50 hypothetical patterns to be tested, which can hardly be qualified as a trivial task for a human analyst.

Analogously, in both KINSHIP and MPD we need to find the "simplest" analysis (in the sense described earlier). I cannot go into details here and will only mention that this reduces to computing a minimal set cover (or converting a Conjunctive Normal Form into Disjunctive Normal Form), a task which is very hard even for computers (strictly, an NP-complete computational problem). In effect, in similarly computationally complex tasks the difference between using and not using a computational tool is not simply a matter of economy or waste of time and efforts, but is rather the difference between getting and not getting a solution at all.

3.2. *The guaranteed consistency of machine results*

Computer discovery programs guarantee the consistency of the results obtained, which is by no means always the case with human solutions. Thus, KINSHIP found errors in componential analyses of kin terms offered in the literature, and MPD discovered inaccuracies in a classical componential analysis of Russian phonemes. A striking result in this vein was found by the system for linguistic universals. From a given collection of data, linguists have attempted to find all and only the universals consistent with these data. From the very same data set, UNIV discovered many more universals than the two most significant works in the field (Greenberg, 1966) and (Hawkins, 1983) have been able to uncover.

3.3. *Portability of machine methods to new tasks and domains*

Computer discovery programs would normally employ general problem solving methods (for classification, generalization, explanation, etc.) that are readily portable to other tasks and domains. MPD, for example, has been used for classification tasks at the different levels of linguistic analysis (semantics, phonology, typology), pattern-recognition, and has further been applied (by Valdés-Pérez) to data from biology, criminology, and musicology. UNIV is capable of discovery of logic patterns whose scientific significance is by no means confined to language universals, or linguistics for that matter. The heuristics MILL models ensure its cross-scientific applicability, though it has not yet been applied outside of linguistics. A most important methodological implication for linguistic research from this is the realization (to my mind, still deplorably lacking) of the unity of endeavours in the different linguistic disciplines, and indeed between linguistics and the other sciences.

3.4. *The potential machines provide for our gaining new insights*

The degree of formalization discovery programs require often results in our deeper understanding of the tasks modelled. Implementing KINSHIP e.g. led us to define explicitly three types of componential analysis, depending on the presence/absence of redundant feature-values in a kin term meaning description ("fully redundant", "partially redundant", and "non-redundant"). Though implicitly present in the studies of the practitioners of the art, they have not been hitherto explicitly recognized. A further theoretical advance, made possible just because of the availability of our system, is related to the old problem of the theoretically possible multiple solutions for kinship systems. Our system, with its capacity to produce alternatives, made conspicuous that not only theoretically, but practically, the number of solutions can be immense, which resulted in our proposing two new coherence constraints normally leading to unique, or just a few, componential analyses.

4. CONCLUSION

Despite the preliminary nature of the reported systems, and the fact that they are concerned basically with data-driven discovery (and not with theory-driven discovery, that has also been important historically in linguistics), the offered arguments, I presume, make a good point for linguistic machine discovery. So linguistics needs to re-consider its present-day negative attitude toward (mechanical) discovery. This new position requires that contemporary linguistics produces both additional historical reconstructions, laying emphasis on the problem solving involved (unfortunately, very scarce to-date), and methodological work recasting some of the traditional linguistic problems in the formal terms of cognitive science and AI. Undoubtedly, to meet these new challenges, new creative efforts would be needed, but once some success is achieved in this direction, computer programs would be much more readily constructable to offer a significant aid to linguists in their future creative endeavours.

ACKNOWLEDGEMENTS

This work was partly supported by the US National Science Foundation Division of International Programs and by the Bulgarian Ministry of Education and Science.

REFERENCES

Cherry, C., M. Halle and R. Jakobson: 1953, Toward the Logical Description of Languages in their Phonemic Aspect. *Language* 29: 34–47.

Chomsky, N.: 1957, *Syntactic Structures*. Mouton: The Hague.

Greenberg, J.: 1966, Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements. In J. Greenberg (ed.), *Universals of Language*. Cambridge, Mass: MIT Press, 58–90.

Hawkins, J.: 1983, *Word Order Universals*. N.Y.: Academic Press.

Hanson, N.: 1958, *Patterns of Discovery*. Cambridge: Cambridge University Press.

Langley, P., H. Simon, P. Bradshaw and J. Zytkow: 1987, *Scientific Discovery: Computational Investigation of the Creative Process*. Cambridge, Mass: MIT Press.

Lounsbury, F.: 1964, The Structural Analysis of Kinship Semantics. In Horace Lunt (ed.), *Proceedings of the 9th International Congress of Linguists*. Mouton: The Hague, 1073–1090.

Mill, J.S.: 1879, *A System of Logic Ratiocinative and Inductive*. London: Longmans Green.

Newell, A. and H. Simon: 1972, *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice Hall, Inc.

Nickles, T. (ed.): 1980a, *Scientific Discovery I*: *Logic and Rationality*. Dordrecht: Reidel.

Nickles, T. (ed.): 1980b, *Scientific Discovery II*: *Case Studies*. Dordrecht: Reidel.

Nogle, L.: 1974, *Method and Theory in Semantics and Cognition of Kinship Terminology*. Mouton: The Hague and Paris.

Pericliev, V.: 1990, On Heuristic Procedures in Linguistics. *Studia Linguistica* 44: 59–69.

Pericliev, V.: 1995, Empirical Discovery in Linguistics. In *Working Notes of the American Association for Artificial Intelligence Spring Symposium Series* “Systematic Methods of Scientific Discovery”. Stanford: Stanford University Press, 68–73.

Pericliev, V.: 1996, Machine Scientific Discovery and the Domain Sciences: Invited Response to “Computer Science Research in Scientific Discovery”. *Knowledge Engineering Review* 11: 67–68.

Pericliev, V. and Raúl Valdés-Pérez: 1998a, A Discovery System for Componential Analysis of Kinship Terminologies. In B. Caron (ed.), *Actes du 16è Congrès International des Linguistes (Paris, 20–25 juillet 1997)*. Published by Pergamon/Elsevier on CD–ROM.

Pericliev, V. and Raúl Valdés-Pérez: 1998b, A Procedure for Multi-class Discrimination and Some Linguistic Applications. In *COLING, Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (August 10–14)*, Montreal, Quebec, Canada, 1036–1042.

Pericliev, V. and Raúl Valdés-Pérez: 1998c, Automatic Componential Analysis of Kinship Vocabularies with a Proposed Structural Solution to the Problem of Multiple Models. *Anthropological Linguistics* 40: 272–317.

Shrager, J. and P. Langley (eds.): 1990, *Computational Models of Scientific Discovery and Theory Formation*. San Mateo, CA: Morgan Kaufmann.

Valdés-Pérez, R. and V. Pericliev: 1997, Maximally Parsimonious Discrimination: A Task from Linguistic Discovery. In *Proceedings of the 14th US National Conference on Artificial Intelligence, AAAI97*. Menlo Park, CA: AAAI Press, 515–520.

Valdés-Pérez, R. and V. Pericliev: 1999, Computer Enumeration of Significant Universals of Kinship Terminology. *Cross-Cultural Research* 33: 162–174.

Institute of Mathematics & Informatics
G. Bonchev Str., bl.8
Bulgarian Academy of Sciences
1113 Sofia, Bulgaria
E-mail: peri@banmatpc.math.acad.bg