PARADIGMS: A COMPUTER MODEL

Michael J. Liebhaber

Child Language Program
University of Kansas

This project grew out of an attempt to model the inflection learning theory proposed by Pinker (1984). The model only captures a portion of the theory, and therefore, this paper is not intended to be a critique of the proposed theory or of Lexical Functional Grammar (see Kaplan & Bresnan, 1982) upon which the theory rests. The purpose of the model was to provide a means for exploring practical and theoretical ideas related to inflection learning specifically and to model building in general. The program is written in Intellect-ul LISP[1] on a SAGE II microcomputer.

The portion of the theory that the model covers describes how children learn inflections for words. An inflection is a morphological alteration of a word that changes that word's grammatical category. For example, the change from present to past tense. The theory modeled here proposes that children, and adults, use paradigms to extract, store, and produce inflections. A paradigm is a notational device for listing of all of the inflections for a particular word.

Pinker uses two types of paradigms, word-specific and general, to represent inflectional information about words. Word-specific paradigms list the entire inflected form of a word while general paradigms only list the inflection, such as in walked versus -ed. General paradigms, then, can be used during language processing to predict the correct inflection for words that share similar characteristics. Recent experimental evidence (Stemberger & MacWhinney, 1986) supports the psychological validity of separate storage for whole inflected words and inflectional affixes.

Assumptions

A major assumption underlying the inflection learning theory is that people use paradigms to encode some aspects of lexical knowledge. Pinker (1984) cites

several reasons for assuming that language learners use paradigms to keep track of inflections. Primary among them is that paradigms can account for developmental phenomena such as restricted initial usage of inflections on words, segmentation errors made by children, and irregular verbs that are learned as easily as regular verbs.

Grammatical Relationships

Another key element of the theory are grammatical relationships. The relationships are assumed to be innate. The model uses a subset of the relationships, shown in Table 1, proposed in the theory.

Table 1

Grammatical relationships used by the computer model

Feature Names	Feature Values
vtype	transitive, intransitive
tense	past, present
aspect	ongoing, completed
number	singular, plural
trole	trans-agent, trans-patient, intrans-actor, goal, beneficiary
object	theme
gender	male, female
person	1st, 2nd, 3rd

Note. Vtype = verb type and trole = thematic role.

Input

Pinker (1984) assumes that children can extract meanings of words and the relationships among them from a stream of input. Additionally, he states that children can recognize grammatical features such as person and tense. Therefore, the words that are input to the model are enriched with information about the word's utterance context and information regarding the thematic roles, such as agent or instrument, of the word.

The argument for including contextual information in the input sentence is that children, with the possible exception of infants and those with severe handicaps, are aware of their surroundings. It is also assumed that children are able to infer meaning from discourse contexts (Pinker, 1981; Schlesinger, 1977; Wexler & Culicover, 1980). Therefore, the contextual information allows the model to be aware of the utterance context. One problem with this argument is that words with similar meanings may not belong in the same paradigm. Misclassifications resulting in defective paradigms are bound to occur. For example, it would be inappropriate to use the past tense inflection for scurry, -ied, on the verb run to produce runned, even though these words are semantically related. Children make this type of error and spontaneously recover from it. Therefore, inflection learning theory must consider the degree of similarity among word meanings and supra-paradigm procedures that watch for and repair defective paradigms. This is clearly one locus of interaction between syntax and semantics.

Lexical Entries

The nature of lexical entries has also been the subject of research (Carey, 1978; Fowler, Napps, & Feldman, 1985; MacKay, 1976; Morton, 1969; Taft & Hambly, 1985). Lexical entries and the amount and kind of information contained in them are at the center of language processing for the model. The problem for this model, and for computer models in general, is to achieve a balance between theory and practicality when specifying the form and structure of lexical entries. That is, how much information is stored in an entry besides the phonological form of a word? Or, at what point does a lexical entry become indistinguishable from other knowledge representations. The program assumes that lexical entries contain the word itself, pointers to the word's features and paradigm, and pointers to semantically related words.

Representation of Linguistic Data

Some of the major variables are shown in Table 2 and are discussed below. They are feature, lexical, and sentence variables. The use of these variables is covered under the discussion of the model's processing. Linguistic data are represented in the program as

individual values, lists of values, and as properties of values and lists. That is, in all forms that LISP allows. Values are data that are assigned to a variable and properties are data that describe features of a variable. For example, a sentence has a list of words as its value, while a lexical entry is defined only by a set of features. The choice between value and property is relatively arbitrary to the program. However, the representations were chosen to correspond with plausible natural representations.

Feature Equations

A feature equation is formed by linking an innate grammatical feature and an input feature value. They are a critical element in the model since they connect innate and environmental features and they are used to build the paradigms for the entry. Feature equations are properties of feature variables. The feature variable names that are used by a particular lexical entry are stored in the feature list of that lexical entry.

Lexical Entries and the Lexicon

A lexical entry is a combination of properties of a lexical entry variable. The properties describe various features of a word: the word itself, its part of speech, strength, and associated feature equations, word-specific paradigm, and context. Part of speech is inferred from natural categories (e.g., words that have thematic role markers are usually nouns, actions are verbs, etc.). Strength is an indication of how often the word has been accessed. A lexical entry can have more than one feature equation. Only equation and paradigm variable names are kept in a lexical entry, not the values. Each lexical entry variable name is stored on a list called the lexicon.

Input Sentence

Sentences are list of lists where each sublist represents the information about a word in the sentence. Each word sublist contains the word, a stress indicator, a feature value, and some context words. Sentence (1) is an example of a two-word intransitive sentence, Billy jumped. Figure 1 is the lexical representation for jumped.

```
(1)  ((Billy 3 (intrans-actor) (friend))
      (jumped 1 (past) (hop over)))
```

Table 2

Names, values, and property lists of major variables in the model

Variable = Value	Property = Value
Fx = none	Feature Equation = (<innate feature name> = <input feature value>)
Lx = none	Name = <word> Part-of-Speech = <NOUN,VERB,?> Strength = <1,2,3,...,100> Feature Equation = (<Fx> ...) Word Specific Paradigm = <Sx> Context = (<input context word> ...)
LEXICON = (L1 L2 L3 ... LN)	none
SENTENCE = (<word1-N>)	none

where <wordX> = (<word> <stress> (<feature value> ...) (<context word> ...))

Note. x is a number generated by LISP when it creates a variable. It starts at 1 and increments sequentially for each new variable.

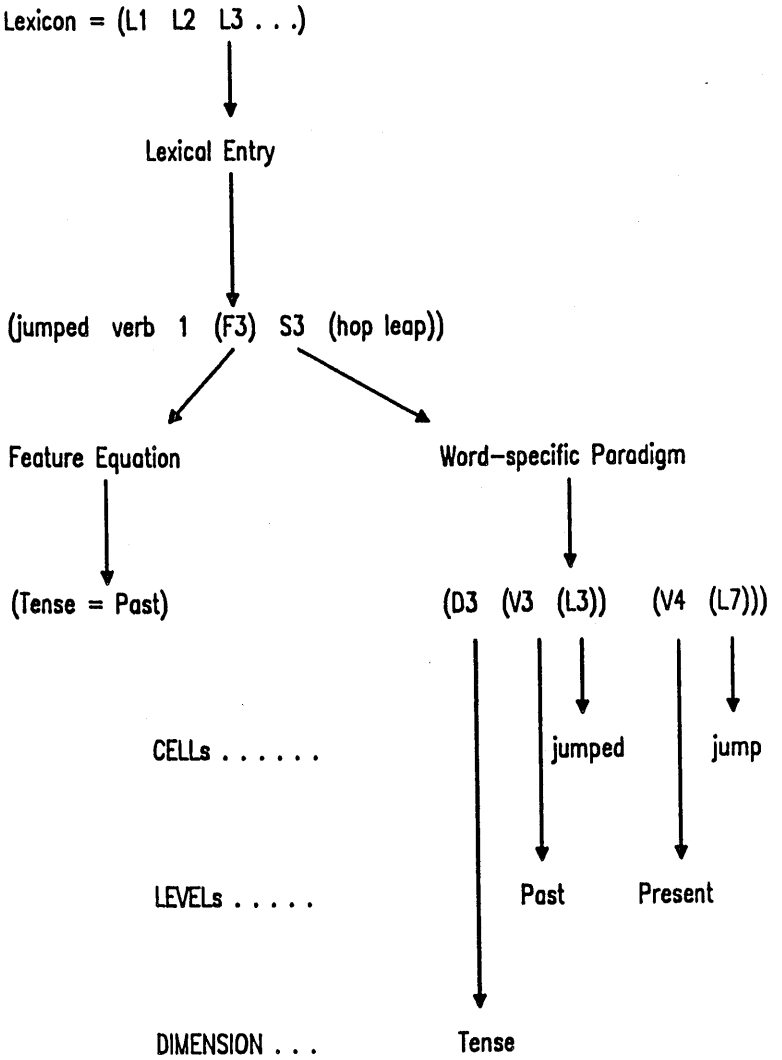

Figure 1. Expanded lexical entry for jumped.

Paradigms

Paradigms are represented in Pinker's model as multidimensional arrays of morphemes. The dimensions of the array are the feature names. Feature values form levels within each dimension. Morphemes are said to be in a cell of the paradigm. The cells are referenced by dimensions, such as TENSE, and levels, such as PAST. These are illustrated in Figure 1. A paradigm must have at least one dimension, each dimension must have at least one level, and each level can have one or more lexical entries.

Paradigm variables are illustrated in Table 3. The program represents paradigms as list of lists. Every dimension of a paradigm is a sublist of the paradigm and every level of a given dimension is a sublist of that dimension list. Finally, the cell is a sublist of the level list and contains lexical entry variable names, Lx. Dimensions and levels only appear once in a paradigm; otherwise they would be redundant, as in (2).

(2) incorrect (TENSE (PRESENT (L2)) (PRESENT (L8)))
correct (TENSE (PRESENT (L2 L8)))

Processing

The program processes sentences word by word from left to right. There is no practical limit other than computer memory restrictions to the kind of words or the number of words per sentence that may be entered. Obviously, there are theoretical constraints on the input such as segmentation, word order, and so on. This model does not "understand" what it is reading in that it does not assign a syntactic structure or semantic interpretation to the input.

Pinker (1984) proposed eleven rules for learning inflections with paradigms. Currently, the program models a subset of the rules dealing with recognizing input features and manipulating word-specific paradigms. Paradigms are formed in the inflection learning theory by (1) selecting a feature value (e.g., PLURAL) from the input utterance, (2) putting the value into a new feature equation, such as NUMBER = PLURAL, which gets added to the lexical entry (LE), (3) transforming the feature equation into a word-specific paradigm (WSP), and (4) removing word stems from the WSP to create a general paradigm (GP).

Table 3

Names, values, and property lists of paradigm variables

Variable = Value	Property = Value
Sx = (<word specific paradigm>)	Stem = <word stem> Change = <0,1,2,...,100> GP = <Gx>
Gx = (<general paradigm>)	none

where <word specific paradigm> =
(<dimension1> (<levelA> (<Lx> ...)) ... (<levelZ> (<Lx> ...))) through
(<dimensionN> (<levelA> (<Lx> ...)) ...)

and <general paradigm> =
(<dimension1> (<levelA> (<affix> ...)) ... (<levelZ> (<affix> ...))) through
(<dimensionN> (<levelA> (<affix> ...)) ...)

Note. x is a number generated by LISP (see Table 2). See the text for a discussion of other variables.

Example Run

The model was given the sentence shown in (3). The example follows the nonword tezam through the model. The result of the run will be a new lexical entry for tezam and revised paradigms.

(3) ((Abby 1 (intrans-actor) (friend))
(tezam 1 (completed) (sleep)))

The program processes sentence (3), one word at a time, using the following seven steps. The first four steps process the word at the level of the lexical entry. Step five places the word in a WSP, and step six forms a GP from the WSP.

1. Match the input feature value, COMPLETED, with its corresponding innate feature name, ASPECT. The match is constrained by the innate grammatical relationships defined in Table 2. If a match cannot be made then enter another value and try matching again, or quit the program. Feature names cannot be entered since they are assumed to be innate.

2. Use the feature name and value to make a feature equation for the word. The resulting equation is: (ASPECT = COMPLETED).

3. Look for the word in the lexicon. If the word can't be found, then create a new LE by generating a new LE variable, filling the new LE with the word, strength, and context from the input. Assign part-of-speech based on input feature value. Assign a WSP to the LE by:
   (1) Use the WSP of a word that is already in the lexicon and which is related semantically, morphologically, or phonologically to the input word.
   (2) If a related word isn't found, then make a new WSP. The WSP is empty and will be filled in later.

   Otherwise, quit the program if a lexical entry wasn't found and a new one wasn't created for some reason.

The new lexical entry is: (TEZAM VERB 1 () S12 (SLEEP))

4. Add the feature equation to the LE only if:
   C1 - it isn't a duplicate of one that is already

attached to the LE, or
C2 - the feature name is in one of the LEs feature equations and it is a thematic role feature.

Constraint C1 prohibits redundant feature equations within a LE, and constraint C2 only allows thematic role features to have multiple equations. For example, a word such as boy can have agent and patient roles, but it cannot be singular and plural since plural requires a different form, boys.

The lexical entry is now: (TEZAM VERB 1 (F9) S12 (SLEEP))

5. Add the lexical entry to its WSP by the following:
   (1) If the feature name IS NOT a dimension of the WSP then add it as a new dimension of the WSP. Add the feature value as a new level in the new dimension. Put the LE into its cell in the new level, replace the old WSP with the new one, and go on to 5.(4).
   (2) If the feature name IS a dimension of the WSP but the feature value IS NOT a level in this dimension then add the feature value as a new level of the dimension. Put the LE into its cell in the new level. Replace the old dimension with the revised dimension, and the old WSP with the new one. Go to 5.(4).
   (3) If the feature name IS a dimension of the WSP and the feature value IS a level in this dimension then replace the LE in the cell with the new LE if:
      C3 - the stress of the new LE is greater than the strength of the existing LE in the cell, or
      C4 - the LE in the cell is pre-emptable.
   Next, put the LE into its cell in the new level. Replace the old level, dimension, and WSP with the revised values. Go to 5.(4).

Constraint C3 is the program's version of the Unique Entry Principle (UEP). The UEP limits the number of words that can occupy a cell in a paradigm: one word in most cases. The program is not flexible when it applies the UEP and the word with the greater strength will always be placed in the cell. A cell entry is said to be pre-emptable, as in C4, when the entry is a prediction made from the affixes in the GP. Predicted entries can be pre-empted by words in

the actual input.

(4) WSP processing is finished. The complete WSP for tezam, and related words, is in Table 4. Go on to the next step, 6.

Table 4

Word-specific paradigm for the example run of the model

	TENSE	Present		Past	
	ASPECT	Ongoing	Completed	Ongoing	Completed
NUMBER	Singular	extezamin	extezam	tezamin	tezam
	Plural	exteamin	exteam	teamin	team

6. If the WSP for the input word has more than one dimension then try and form a GP with the following steps.

(1) If the WSP already has a GP associated with it then retrieve the stem of the word.
C5 - If the stem hasn't changed recently then:
(a) Form the affix for the word by removing the stem from the word.
(b) Retrieve the LE from the cell, at the appropriate dimension and level, of the GP.
(c) If there isn't a LE to retrieve then add the affix to the GP and go to step 7.
(d) If the new and old affixes are identical then replacing the existing LE would be redundant so go to Step 7.
(e) If this point is reached then the old and new affixes conflict so the entire WSP must be reanalyzed. Go to the next step, 6.(2).

(2) This WSP does not have a GP associated with it or it must be reanalyzed. Make a new GP

variable if necessary. Attach its name to the property list of the WSP (see Table 3). This will associate the WSP with the GP.

(3) Only abstract a WSP into a GP if:
C6 - The WSP has more than one dimension, or
C7 - The WSP has one dimension, but has more than one level in that dimension.
Otherwise, go to step 7.

Constraints C6 and C7 exclude one word paradigms from GP analysis. They correspond to rule I9(c) in the theory (Pinker, 1984:198). The next step, 6.(4), forms the GP from the WSP and is illustrated in Table 5 for the TENSE dimension.

Table 5

Affix and stem extraction for the TENSE dimension in the example run the model

	Present				Past			
	extezamin	extezam	exteamin	exteam	tezamin	tezam	teamin	team
Common	exteam				team			
Affixes	ex-				0-			
Stem	team							

(4) Perform the following for each dimension in the WSP:
(a) For each level in the current dimension; extract the string of letters that are common to all of the words in this level. This will give one "word" for each level in this dimension.

(b) Look through each of the extracted letter strings for the STEM and AFFIXes. There will be one stem and one or more affixes (one affix for each level). The STEM of this word is the word formed from the letters that are common to all of the strings. The AFFIXes are the portions of the strings that are unique to that string.

General paradigm processing is now complete. The entire GP for the WSP in Table 4 is:

((TENSE (Present (ex-)) (Past (O-)))
(ASPECT (Ongoing (-in)) (Completed (-O)))
(NUMBER (Singular (-z-)) (Plural (-O-))))

7. The processing for the current word is completed. If there are more words in the sentence then get the next word in the sentence. Return to step 3 and process the word. Otherwise, the INFLECT program is finished.

Discussion

The model deviated from the rules proposed in the theory in three ways. First, formation of feature equations is driven by features in the input rather than sampling from innate features (see step 1.). This was done to eliminate the need for resampling the innate feature list until a correct match was found. Second, the model assumes that stems of words are not known. This posed some difficulties for GP abstraction, in step 6, which are discussed below. Third, conflicting entries in a GP cell cause the entire GP to be reanalyzed to ensure that all of the affixes are correct (step 6.(1)(e)). The unique entry principle and pre-emptability are employed in the theory to remove conflicts. These methods assume that other entries in the GP are unaffected by the new word causing the conflict.

Theoretical Considerations

Theoretical problems that need to be resolved include the selection of word-specific paradigms, explicating suppletive handling procedures, and

circumventing problems caused by lack of data.

Selection of word-specific paradigms. The selection of the correct WSP for new words is not addressed by the theory. The program, for a given input word, searches context lists of all lexical entries for words that match the input context. If one is found the program assigns the WSP to the new word. In other words, the program performs a semantic search. Searches along the morphological and phonological dimensions of a word can also be made. These are acknovledged, but not implemented, in the program. Pinker (1984:194) also acknovledges the importance of morphological and phonological information in WSP assignment. The sequential logic and interpretation of these matches will be difficult to implement. For example, what match should be done first, and what does it mean when one match is made, say phonological, but not any others?

An additional consideration is the degree of similarity required before a match can be made. The exactness of the match is not controlled in the program so the selected lexical entry may or may not be appropriate. This may be satisfactory initially, and it does account for semantic overextension errors such as the one made by the beginning reader in (4). However, some constraints must eventually be placed on the search and match procedure.

(4) Mother: What are you reading?
Child: A story about keeping water.
M: Keeping water?
C: Yeah, around the house.
M: What's the title of the story?
C: Ways to Save Water at Home.

Suppletion. Suppletion became a problem for the model when GPs were being abstracted from WSPs. The program can handle suppletive forms such as in (5), up to a point. Since the words within each pair are semantically similar, they will be placed in the same WSP. However, the model relies on morphology for abstraction and will give the results shown in (5). The "0" indicates no stem, which is correct since the program does not know that it can arbitrarily consider the present tense morpheme as the stem.

(5) go and vent --> STEM = 0 AFFIXES = go vent
run and ran --> STEM = rn AFFIXES = -u- -a-

Suppletive forms only cause problems when they reach the level of GP abstraction. One method for avoiding the problems illustrated in (5) is to limit the application of the abstraction process to WSPs that have a sufficient number of lexical entries. This is what Pinker (1984:198) does in rule I9(c). However, this raises the question of what is a sufficient number. The program requires at least two words in a dimension of a WSP before it will attempt to form a GP from that WSP. Another method for avoiding problems, which is also used by Pinker, is to assume that word stems are known before GP abstraction is attempted. However, stems must be abstracted at some point in language development, and assuming their existence avoids learnability issues. For example, are stems actually learned through abstraction, or is one cell in a paradigm innately recognized as the stem?

Insufficient data to create general paradigms. Suppletion touched on another issue in paradigm usage. Namely, how is GP abstraction handled when there are only a few words in the WSP undergoing abstraction? The correctness of stems and affixes of verbs depends on the completeness of the WSP. The program reanalyzes a WSP every time one is accessed. Reanalysis could be limited by tracking the number of times a stem has changed relative to the number of times the WSP it appears in is accessed (constraint C5) and by avoiding WSPs with too few entries (as in the theory & in constraints C6 & C7).

Meta-Paradigms. Many of the problems discussed above could be avoided by allowing general paradigms to communicate with each other. This issue was not addressed in the theory or in the model. Interparadigm communication would let paradigms compare information, such as affixes, and would presumably circumvent problems caused by lack of data within one paradigm.

Conclusion

The program presented in this paper is an initial attempt to model the acquisition and use of inflectional paradigms. The model successfully builds feature equations and limited word-specific paradigms. General

paradigms were discussed, but were not implemented in the model. Additional consideration must be given to the form of paradigm representation and to inter-paradigm relationships before the model can be extended to include general paradigms. Specifically, three questions must be addressed. First, how are word-specific paradigms found for novel words? This becomes a question of what makes words similar? Second, what constrains rule application, as in the case or regular and suppletive morphology? And finally, how do paradigms interact? The answers to these questions are not settled and depend on ones theoretical orientation. However, they must be addressed in order for a complete, testable theory to be developed.

NOTES

[1]Intellect-ul LISP is a trademark of PCD Systems. The source code for the model is available from the author. SAGE II is a trademark of SageMicro of Reno, NV. The SAGE II has a Motorola 68000 cpu, 256K of RAM, and runs the UCSD P-system operating system.

REFERENCES

Carey, S. (1978). The child as word learner. In M. Halle, J. Bresnan, & G. A. Miller (Eds.), Linguistic theory and psychological reality (pp. 264-293). Cambridge, MA: The MIT Press.

Fowler, C. A., Napps, S. E., & Feldman, L. (1985). Relations among regular and irregular morphologically related words in the lexicon as revealed by repetition priming. Memory & Cognition, 13, 241-255.

Kaplan, R. M., & Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), The mental representation of grammatical relations (pp. 173-281). Cambridge, MA: The MIT Press.

MacKay, D. G. (1976). On the retrieval and lexical structure of verbs. Journal of Verbal Learning & Verbal Behavior, 15, 169-182.

Morton, J. (1969). Interaction of information in word recognition. Psychological Review, 76, 165-178.

Pinker, S. (1981). Comments on the paper by Wexler. In C. L. Baker & J. J. McCarthy (Eds.), The logical problem of language acquisition. Cambridge, MA: The MIT Press.

Pinker, S. (1984). Language learnability and language development. Cambridge, MA: Harvard University Press.

Schlesinger, I. M. (1977). The role of cognitive development and linguistic input in language development. Journal of Child Language, 4, 153-169.

Stemberger, J. P., & MacWhinney, B. (1986). Frequency and the storage of regularly inflected forms. Memory & Cognition, 14, 17-26.

Taft, M., & Hambly, G. (1985). The influence of orthography on phonological representations in the lexicon. Journal of Memory & Language, 24, 320-335.

Wexler, K. & Culicover, P. (1980). Formal principles of language acquisition. Cambridge, MA: The MIT Press.