Verb Concatenation in Hmong Njua
A syntactic description
and
its treatment in natural language processing

Dr.Bettina Harrichausen-Mühlbauer
Scientific Center
IBM Germany
Institute for Knowledge Based Systems
Wilckensstr. 1a
6900 Heidelberg
Germany
e-mail: HARRIEHA at DHDIBM1.BITNET
phone: +49 (0)6221-404216

Abstract

Hmong Njua, a language mainly spoken in the highlands of Laos, is a language without inflection, case and agreement system, tense, number and gender markings. It uses neither derivational nor inflectional morphology. Its strategy of forming sentences is, by and large, similar to that of many other East Asian languages of the isolating type: sentences are formed by concatenating nouns and verbs, and grammatical words seldom occur.
A linguist studying a language such as Hong Njua is, thus, faced with the challenge of describing its multi-verb-constructions: What are the semantic relations between verbs? What are the constituent structures, or the lack of it, in those constructions? Do those concatenated verbs constitute one proposition or a complex or compound structure?
In this paper, the phenomenon of verb-concatenation in Hmong Njua is described in detail.

Furthermore, the PLNLP natural language processing system is described, demonstrating how verb-concatenations can be processed computationally, including the possibility of detecting syntactic errors, such as the incorrect verb-ordering in a concatenated verbal structure.

Verb-Concatenation

Verb-concatenation, a well-known and well-distributed feature among Southeast-Asian languages, describes also a characteristic phenomenon in Hmong Njua.

When speaking of verb-concatenation, several similar terms are used in the literature, which might lead to confusion: ***multi-verb construction***, ***verb-concatenation***, and ***serial verbs***. Although all these structures share many features, a clear distinction is necessary:

- ***Multi-verb constructions*** contain more than one verb in a sentential structure. Two types of multi-verb constructions can be distinguished: those with one predication, i.e. sentences of the form SVO or SOV, and those with more than one predication, i.e. coordinated structures.
- By ***verb-concatenation***, we understand the phenomenon of combining verbs in a sequence without interruption of other word-classes:

  V + V + V + V + V[1]
- ***Verb-series***, or serial verbs, describe the linear combination of verbs with possible interruptions of the sequence:

  V1 + OBJ1 + V2 + OBJ2

According to these definitions, this paper will describe verb-concatenation of multi-verb constructions with one predicate.

While analyzing the phenomenon of verb-concatenation, certain questions will arise and will need to be answered:

1. Which elements or morphemes are modified or modify other elements?
2. Which elements are words of a word-class and which are particles?
3. Which constructions are lexical compounds, i.e. have to be analyzed morphologically, and which are syntactic constructions?

In order to explain this phenomenon, a differentiation into the following 3 main categories or types of verbs is necessary:

1. Verbs of motion (MOTION)
2. Verbs of direction (DIR)
3. Verbs of deictic nature (DEICT)

---

[1] According to Matisoff (1973) this is a structure mainly found in SOV-languages.

Verbclasses

Verbs of motion describe a certain motion by focussing on the kind of motion.

dhla	to run/jump
yaa	to fly
dlaum	to climb
nkaag	to crawl
moog	to go

Fig.1: Examples for verbs of motion

Verbs of direction refer to the direction of a motion, like 'up', 'down', 'into', or 'out of', without any reference to the kind of motion. The meaning of these verbs change according to their use as full verbs, i.e. when they occur alone, or as verbs of direction, i.e. when used as complements to the main verb in a concatenate structure.

	full-verb	concatenation
	---	---
nce	to climb, ascend	up
ncig	to circle	around
hlaa	to cross	across
nqeg	to descend	down
nkaag	to enter	into

Fig.2: Examples for verbs of direction

The third class, that of ***deictic verbs***, can be regarded as a subclass of either verbs of motion or verbs of direction, as their meaning implies **motion** as well as **direction of a motion** with reference to the speaker.
But as this third class, which only consists of the 2 verbs:,

lug	to come / here
moog	to go / away

Fig.3: The closed class of deictic verbs

carries certain very special features, it gets treated as a separate class. Again, as with verbs of direction, these verbs can be used alone (then they carry their full-verb meaning) or in a concatenated structure, where they carry a meaning which can be compared with the English adverbs **away** and **here**.

Deictic verbs in Hmong Njua and separable prefixes in German

While analyzing and describing syntactic structures and features of a language, we often compare these structures with similar phenomena in other, usually linguistically related, languages.

During the analysis of Hmong Njua, many features were found that Hmong Njua shares with other Southeast-Asian languages, such as Thai, Burmese, Lahu, or Chinese. Surprisingly, an interesting parallel could be drawn to "deictic particles" in German, with one major difference, that the German deictic particles have already lost the possibility of occurring alone, i.e. they can only be used in combination with a main verb as separable prefixes. Just as in Hmong Njua, these prefixes point to the direction of a motion with reference to the speaker:

```
herkommen (to come here)          : Er kommt  her. (He comes here.)
                                      MOTION DEICT

herüberkommen (to come over here) : Er kommt  her-  über. (He comes over here.)
                                      MOTION DEICT DIR

hingehen (to go there/away)       : Er geht   hin. (He goes away.)
                                      MOTION DEICT

hinübergehen (to go away/across)  : Er geht   hin-  über. (He goes away/across).
                                      MOTION DEICT DIR
```

Fig.4: German separable prefixes with deictic implication

Of course this does not imply that Hmong Njua and German are linguistically or typologically related, but such comparisons and parallels can be used as helpful "devices" in natural language processing, as parallel structures can be treated identically by the rules.

Multi-verb constructions with a single predicate

Multi-verb constructions with a single predicate allow 3 different interpretations:

1. They form a compound, i.e. 2 or more verbs form a semantic and syntactic unit, which cannot be separated by other words or particles.
2. According to Li/Thompson (1981), they can function as "serials":
   "A construction that refers to a sentence that contains two or more verb phrases or clauses juxtaposed without any marker indicating what the relationship is between

them, therefore sentences may have the same form but they convey different types of messages because of the meanings of the verbs involved and the relationships that are understood between them".

3. When 2 verbs occur together in a sentence, and the second verb modifies its grammatical category to that of a modifier or complement of the preceding verb. This phenomenon, in which the directional and deictic verbs lose their full-verb meaning and change it to that of a directional or deictic complement respectively, when they follow a verb of motion, will be demonstrated below. 4 combinations are possible:
   a. MOTION + DEICT
   b. DIR + DEICT
   c. MOTION + DIR
   d. MOTION + DIR + DEICT

   Let us first look at examples of each combination before demonstrating how these constructions can be processed by a natural language processing system.[2]

```
MOTION + DEICT
--------------
(1)  Nwg  dlha  moog/lug.
     3sg  run   go  /come
     He/she runs away/here.

DIR + DEICT
-----------
(2)  Nwg  nce    moog/lug.
     3sg  climb  go  /come
     He/she climbs up - and away from the speaker/towards the speaker.

MOTION + DIR
------------
(3)  Nwg  dlha  nkhaa  lub  hoob.
     3sg  run   enter  CL   room
     He/she runs into the room.

MOTION + DIR + DEICT
--------------------
(4)  Puav  dlha  tawm   moog.
     3pl   run   leave  go
     They ran out - and away from the speaker
```

Fig.5: Examples for different types of multi-verb constructions

---

[2] For more examples see Harriehausen (1989a), chapter 5.3.3.2.

Those examples demonstrate that the maximal number of verbs in sequence are 3.[3] This concatenation can be extended by verbal particles and modal-/auxiliary verbs. The discussion, whether such morphemes should be included in the category of full-verbs, i.e. whether they should be counted as components in a concatenated verb-sequence, would go beyond the scope of this paper, but can be read in Harriehausen (1989a, pp.166 ff).

Verb concatenations cannot be interrupted by negation, temporal-, or aspectual markers. Such makers[4] have to precede or follow the entire verbal sequence and cover the complete concatenation:

(5a) Puab tsi dhla tawm lug.
3pl NEG run leave come
They didn't come running out.

(5b) * Puab dlha tsi tawm lug.
(5c) * Puab dlha tawm tsi lug.

(6) Nwg maam dhla moog.
3sg FUT run go
He/she will run away.

(7) Nwg tau dlha moog.
3sg PST run go
He/she ran away.

(8) Nwg dlha moog lawm.
3sg run go COMPL
He/She ran away.

Fig.6: Sample sentences that demonstrate the strong unit of the verbal structure

These examples show that concatenated structures are very strong units. The only wordclass that can (sometimes) interrupt a verb-concatenation is the noun. Although such construction would resemble a serial-verb-construction, as defined above, we still need to call it a concatenated structure, as

---

[3] The maximal number of concatenated verbs in most Sino-Tibetan languages is 5.

[4] NEG = negation, FUT = future, PST = past, COMPL = completive aspect

- the strong unit of the verbal sequence is still present, i.e. the second (or third) verb doesn't function as a full verb, but carries the meaning of a directional or deictic complement or modifier to the first (main) verb, and
- objects, which follow a verb of motion, are optional, i.e. in most cases, the reference to the object is implied by the nature of the verb.

(9a) Nwg tsaa lub tsheb moog.
3sg drive CL car go
He/She drove the car away.

(9b) * Nwg tsaa moog lub tsheb.

(10a) Nwg thaws rooj ncig.
3sg move chair circle.
He/she moves the chair around.

(10b)* Nwg thaws ncig rooj.

Fig.7: Sample sentences in which concatenations are interrupted by nouns

No example could be found which allows a different ordering of the verbs, i.e.
* DIR + MOTION or
* DEICT + MOTION.

(11a) Nwg ua-luam-dlej hlaa tug haav-dlej.
3sg swim cross CL river
MOTION DIR
He/She swims across the river

(11b)* Nwg hlaa ua-luam-dlej tug haav-dlej.
DIR MOTION

(11c)* Nwg hlaa tug haav-dlej ua-luam-dlej.
DIR MOTION

(11d) Nwg hlaa tug haav-dlej hab ua-luam-dlej.
He/She cross CL river and swim
DIR MOTION

Fig.8: Sample sentences with and without the coordinator 'hab'

The strict order of the verbs can only be reversed by inserting the conjunction hab ("and") between the verb-phrases - as demonstrated in example (11d) -, thus turning

the verb-concatenation into a verb-coordination.[5] The strict order of the verb classes can be explained by the iconic representation of the real world events, i.e. by looking at example (11), it is only logical to swim before crossing the river, and not the other way around. With this strict verb-sequencing, Hmong Njua differs from other Southeast-Asian languages, such as Lahu.[6]

Verb-concatenation in natural language processing

The System

The system we are using for both analysis and generation of natural language sentences has been developed by Heidorn (1972) and has been implemented in various NLP (= natural language processing) projects in IBM worldwide.

Both components are implemented in PLNLP (= Programming Language for Natural Language Processing, pronounced 'Penelope'), which gets translated into Fortran, LISP, and C and runs under VM on the mainframe and (the C-version) under OS/2 on the PS/2.

For the analysis, the parser operates bottom-up, left-right, and parallel, the generation rules are processed top-down, left-right, and serial. Both types of rules are augmented phrase-structure rules. The rules for analysis, or so-called **decoding-rules**, have one or more symbols (both terminal and non-terminal) on the left and non-terminal symbols on the right. Stripped of their augmentations, these rules look like context-free phrase-structure rules:

```
V        -> VP
VP  NP   -> VP
VP  VP   -> VP
```

Fig.9: Samples of the core of decoding rules

---

[5] One could predict that the ungrammaticality of (11b) would be of pure syntactic nature, because the transitive verb 'cross' needs an object, whereas the intransitive verb 'swim' doesn't. However, (11c) still remains ungrammatical.

[6] Examples can be found in Matisoff (1973).

Most of the rules are of the binary type, mainly for the reason of effectiveness. This reason will become apparent later by looking at examples of how verb-concatenations get processed.

The dictionary used carries information of the word-class and morphological-, syntactic-, and semantic features in linear order:[7]

```
nce(VERB P1 P2 P3 P4 P5 P6 HMONG DIR (TRANSL climb))
dlha(VERB P1 P2 P3 P4 P5 P6 HMONG MOTION (TRANSL run))
lug(VERB P1 P2 P3 P4 P5 P6 HMONG DEICT (TRANSL come))
moog(VERB P1 P2 P3 P4 P5 P6 HMONG DEICT (TRANSL go))
nwg(PRON P3 PERS)
```

Fig.10: Sample dictionary entries for Hmong Njua lexemes

Advantages of the PLNLP system

Before looking at Hmong Njua parsing examples, let me point out a few advantages of this NLP-system:

Fitted Parse

The first advantage, that of the so-called ***fitted parse***, is that the system always attempts to return a processed structure (according to a pre-defined order) in cases where a full, complete parse on sentence-level couldn't be reached.
Reasons for such parse failure can be:

- Mistakes in the grammar.
- Grammatical mistakes in the input string.
- Fragments, such as greetings, farewells, formulaic utterances, and also vocatives and extreme ellipses.

After failing to interpret the input string as a complete sentence, the results are turned over to a set of rules which handle the **fitting routine**. This routine tries to find the most

---

[7] P1-P6 : morphological features : P1 = 1st person singular, P6 = 3rd person plural
These features are needed, as this system was built for German and has to check for agreement. Of course these features are meaningless for Hmong Njua, as it is a language without inflectional or derivational morphology.
HMONG : special marker for the language
PERS : personal pronoun
TRANSL : translation

plausible constituent (substring) among the many records which have been produced in the course of the bottom-up parsing process.
The advantage of seeing such a "partial result" - and not simply the message "parse failed" - is very important for the grammar writer for debugging and enhancing the rules. Examples of fitted parses will be given below when demonstrating how concatenated verbs get processed on phrase-level.

Error Correction

Apart from giving "fitted" analyses of input strings, the PLNLP system provides a flag (which can be thought of as a "switch"), which allows the rules to be less restrictive, or "relaxed", and which leads to corrections of the input. By default, this flag is set to NIL, i.e. turned off, resulting in the fitted parse.

In order to be able to correct deviant input, we need to foresee and implement expected errors. Errors of different types, e.g. syntactic or stylistic, are implemented in different sections of the system; i.e. true syntactic errors, such as subject-verb-agreement in Germanic languages or wrong verb-type order in concatenations in Hmong Njua, are implemented in the core rules and can lead to a parse failure, whereas stylistic errors, e.g. using the indicative instead of the subjunctive mood in subordinate clauses in Germanic languages, are implemented as procedures in a separate set of rules which get processed after the core grammar. These types of errors never lead to a parse failure, they only trigger an error message. Again, examples will be given below.

The following illustration shows different sections of the entire system and makes the order of processing visible.

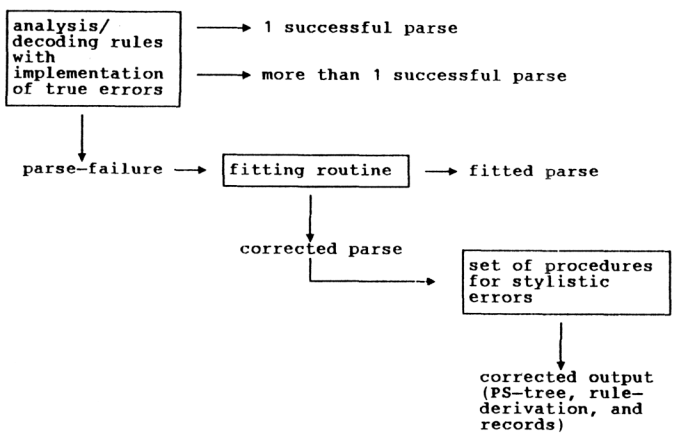

Fig.11: The decoding section of the system

Processing verb-concatenations

Regarding verb-concatenation in Hmong Njua, it will be necessary for the computer to accept the 4 grammatical combinations of the 3 verb-types and reject all other combinations, or at least pinpoint the error. Let us now look at examples:[8]

1. MOTION + DEICT

```
 Enter a sentence or a PLNLP command:
dlha moog.
------------------------------------------------------------
XXXX1  VP1*     VERB1*  "dlha" (run)
               VP2      VERB2*  "moog" (go)
       PUNC1    "."
------------------------------------------------------------
```

Fig.12: Fitted parse for concatenation "dhla moog" (run away)

Fig. 12 shows the phrase-structure tree for the MOTION-DEICT- concatenation "dlha moog" ("run away"). It is a fitted parse (marked by XXXX1), as sentence-level couldn't be reached during processing.

Another display, that of Fig. 13, shows the rule-hierarchy and -derivation for the verb concatenation, e.g. rule number (2581) combines the VP dlha and the VP moog. Such display is important and helpful for the grammar writer during development and debugging of the rules:

```
 Enter a sentence or a PLNLP command:
(PRTREER vp1)

VP1 (2581)
   VP3 (2010)
      VERB1 "dlha" (run)
   VP2 (2010)
      VERB2 "moog" (go)
```

Fig.13: Rule hierarchy of verb-concatenation

---

[8] For better understanding, the English translations of the lexemes are given in brackets.

During parallel processing, many records have been computed. The one covering the entire structure is shown below:

```
Enter a sentence or a PLNLP command:
(PRTREC vp1)
     SEGTYPE    'VP'
     STR        " dlha moog"
     RULES      2010 2581
     RULE       2581 VP3 VP2
     COPYOF     VP3 "dlha" 'DLHA'
     BASE       'DLHA'
     DICT       'dlha'
     INDIC      P1 P2 P3 P4 P5 P6
     HEAD       VERB1 "dlha" 'DLHA'
     PSMODS     VP2 "moog" 'MOOG'
     HMONG      1
     MOTION     1
     TRANSL     'RUN'
     PRED       'DLHA'
     FIT        1
```

Fig.14: Attribute-value-record for verb-concatenation

The left side of the record displays the attributes, such as SEGTYPE, STR, or RULES, and the right side displays the values, which can be of type pointer (such as VP3 or VP2, i.e. pointing to another complete record), number (such as 2010 2581, i.e. naming the rule-numbers involved during processing), and string (such as 'dlha', i.e. words). The main information kept in each record can be interpreted the following way:

- BASE points to the lemma of the head,
- PSMODS (= post-modifier) contains the list of elements, which follow the head.[9]
- RULES and RULE includes the rule derivation.
- INDIC contains the features that get returned from the dictionary look-up.
- TRANSL gives information about the translation of the entry
- MOTION tells us that the head of the structure, i.e. dlha ,is a verb of motion.

---

[9] Although the PLNLP formalism and grammar doesn't strictly follow any of the existing, well-known linguistic theories in computational linguistics, such as GB, GPSG, HPSG, or LFG, it can be compared with dependency-oriented approaches in that the grammar builds its records accordingly, i.e. a head constitutes the "core" of the record with PRMODS (pre-modifiers) and PSMODS (post-modifiers) preceding or following the head respectively.

Analogous to the preceding example, the parallel MOTION + DEICT- structure "dlha lug" ("run towards the speaker") gets processed identically, i.e. the same rules get applied in the same order:

```
 Enter a sentence or a PLNLP command:
dlha lug.
----------------------------------------
XXXX1  VP1*    VERB1*  "dlha" (run)
               VP2     VERB2*  "lug" (come)
       PUNC1   "."
----------------------------------------

 Enter a sentence or a PLNLP command:
(PRTREER vp1)

VP1 (2581)
   VP3 (2010)
      VERB1 "dlha" (run)
   VP2 (2010)
      VERB2 (2003)
         VERB3 "lug" (come)

 Enter a sentence or a PLNLP command:
(PRTREC vp1)
     SEGTYPE   'VP'
     STR       " dlha lug"
     RULES     2010 2581
     RULE      2581 VP3 VP2
     COPYOF    VP3 "dlha" 'DLHA'
     BASE      'DLHA'
     DICT      'dlha'
     INDIC     P1 P2 P3 P4 P5 P6
     HEAD      VERB1 "dlha" 'DLHA'
     PSMODS    VP2 "lug" 'LUG'
     HMONG     1
     MOTION    1
     TRANSL    'RUN'
     PRED      'DLHA'
     FIT       1
```

Fig.15: Fitted parse, rule derivation, and attribute-value-record for "dlha lug" (run here

Of course, verbs can also be processed in isolation. Similar structures with equally complex record-structures are the result:

```
 Enter a sentence or a PLNLP command:
lug.
---------------------------------
XXXX1  VP1*    VERB1*  "lug" (come)
       PUNC1   "."
---------------------------------

 Enter a sentence or a PLNLP command:
(PRTREC vp1)
     SEGTYPE   'VP'
     STR       " lug"
     RULES     2010
     RULE      2010 VERB1
     COPYOF    VERB1 "lug" 'LUG'
     BASE      'LUG'
     DICT      'lug'
     INDIC     P1 P2 P3 P4 P5 P6
     HEAD      VERB1 "lug" 'LUG'
     HMONG     1
     DEICT     1
     TRANSL    'COME'
     PRED      'LUG'
     FIT       1

 Enter a sentence or a PLNLP command:
moog.
---------------------------------
XXXX1  VP1*    VERB1*  "moog" (go)
       PUNC1   "."
---------------------------------
```

```
Enter a sentence or a PLNLP command:
(PRTREC vp1)
    SEGTYPE   'VP'
    STR       " moog"
    RULES     2010
    RULE      2010 VERB1
    COPYOF    VERB1 "moog" 'MOOG'
    BASE      'MOOG'
    DICT      'moog'
    INDIC     P1 P2 P3 P4 P5 P6
    HEAD      VERB1 "moog" 'MOOG'
    HMONG     1
    DEICT     1
    TRANSL    'GO'
    PRED      'MOOG'
    FIT       1
```

Fig.16: Fitted parses and records for the verbs "lug" (come) and "moog" (go)

2. DIR + DEICT

The next parsing example, Fig.18, that of the DIR + DEICT- verb-concatenation nce moog ("to climb away"), shows that processing is identical to the MOTION + DEICT example given above, with the only difference that a different rule, namely rule number 2583, conjoined the VP nce and the VP moog and the record contains the attribute DIR, i.e. information that the head of the structure is a directional verb which can be translated as "climb". Apart from these minor details, the computed structures are alike.

```
Enter a sentence or a PLNLP command:
nce moog.
----------------------------------------
XXXX1  VP1*    VERB1*  "nce" (climb)
               VP2     VERB2*  "moog" (go)
       PUNC1   "."
----------------------------------------

Enter a sentence or a PLNLP command:
(PRTREER vp1)

VP1 (2583)
   VP3 (2010)
      VERB1 "nce" (climb)
   VP2 (2010)
      VERB2 "moog" (go)
```

```
 Enter a sentence or a PLNLP command:
(PRTREC vp1)
     SEGTYPE   'VP'
     STR       " nce moog"
     RULES     2010 2583
     RULE      2583 VP3 VP2
     COPYOF    VP3 "nce" 'NCE'
     BASE      'NCE'
     DICT      'nce'
     INDIC     P1 P2 P3 P4 P5 P6
     HEAD      VERB1 "nce" 'NCE'
     PSMODS    VP2 "moog" 'MOOG'
     HMONG     1
     DIR       1
     TRANSL    'CLIMB'
     PRED      'NCE'
     FIT       1
```

Fig.17: Fitted parse, rule derivation, and record-structure for DIR-DEICT-verb-concatenationsentenc

3. MOTION + DIR

The third possible ordering of verbs, that of MOTION + DIR, is demonstrated by the verb-concatenation "dlha nce" ("run up"):

```
 Enter a sentence or a PLNLP command:
dlha nce.
----------------------------------------
XXXX1  VP1*    VERB1*  "dlha" (run)
               VP2     VERB2*  "nce" (climb)
       PUNC1   "."
----------------------------------------

 Enter a sentence or a PLNLP command:
(PRTREER vp1)

VP1 (2582)
   VP3 (2010)
      VERB1 "dlha" (run)
   VP2 (2010)
      VERB2 "nce" (climb)
```

```
Enter a sentence or a PLNLP command:
(PRTREC vp1)
     SEGTYPE   'VP'
     STR       " dlha nce"
     RULES     2010 2582
     RULE      2582 VP3 VP2
     COPYOF    VP3 "dlha" 'DLHA'
     BASE      'DLHA'
     DICT      'dlha'
     INDIC     P1 P2 P3 P4 P5 P6
     HEAD      VERB1 "dlha" 'DLHA'
     PSMODS    VP2 "nce" 'NCE'
     HMONG     1
     MOTION    1
     TRANSL    'RUN'
     PRED      'DLHA'
     FIT       1
```

Fig.18: Fitted parse, rule derivation, and record-structure for MOTION-DIR verb-concatenat

Again, as can be expected by now, processing is identical with the only difference of a separate rule (2582) conjoining the two verbs.

At this point one might ask why separate rules were written when these structures are so much alike. This is a question the grammar writer faces constantly and has to choose between the options of

- writing separate rules for similar constructions, i.e. keeping the rules themselves fairly simple, but making the whole grammar more complex by adding more rules (and thus also running the risk of multiple parses) and

- writing one complex rule which needs to be rather complex in itself, i.e. needs to contain many conditions/augmentations, that allow or block certain structures.

If we wrote one single rule for the examples above, i.e.

```
VP (MOTION or DIR) + VP (DIR or DEICT)  -->  VP ...  ,
```

we would need to include a condition which blocks DIR + DIR.
In many cases, either choice is possible and a lot of times it's up to the "taste" of the grammar writer how he/she wants certain structures to get processed. In general one

might want to prefer the second option, i.e. writing fewer, but more complex, rules, in order to avoid the "danger" of multiple parses.

4. MOTION + DIR + DEICT

Last not least, the most complex verb-concatenation, that of MOTION + DIR + DEICT, is shown in Fig.19:

```
 Enter a sentence or a PLNLP command:
nwg dlha nce moog.
--------------------------------------
DECL1  PRON1   "nwg" (he/she/it)
       VERB1*  "dlha" (run)
       VP3     VERB2*  "nce" (climb)
       VP2     VERB3*  "moog" (go)
       PUNC1   "."
--------------------------------------

 Enter a sentence or a PLNLP command:
(PRTREER 1)

DECL1 (4000)
   SNTBEG1 ""
   VP7 (2584)
      PRON1 "nwg" (he/she/it)
      VP8 (2581)
         VP9 (2582)
            VP6 (2010)
               VERB1 "dlha" (run)
            VP3 (2010)
               VERB2 "nce" (climb)
         VP2 (2010)
            VERB3 "moog" (go)
   PUNC1 "."
```

```
Enter a sentence or a PLNLP command:
(PRTREC 1)
    SEGTYPE   'SENT'
    SEGTYP2   'DECL'
    STR       " nwg dlha nce moog ."
    RULES     2010 2582 2581 2584 4000
    RULE      4000 SNTBEG1 VP7 PUNC1
    COPYOF    VP7 "nwg dlha nce moog" 'DLHA'
    BASE      'DLHA'
    DICT      'dlha'
    INDIC     P1 P2 P3 P4 P5 P6
    PRMODS    PRON1 "nwg" 'NWG'
    HEAD      VERB1 "dlha" 'DLHA'
    PSMODS    VP3 "nce" 'NCE'
    PSMODS    VP2 "moog" 'MOOG'
    PSMODS    PUNC1 "." '.'
    SUBJECT   PRON1 "nwg" 'NWG'
    HMONG     1
    MOTION    1
    TRANSL    'RUN'
```

Fig.19: PS-tree, rule-derivation, and record for MOTION+DIR+DEICT verb-concatenation

Fig.19 displays a sentential structure, i.e. a successful (not fitted-) parse. DECL1 tells us that the parsed sentence is of declarative mode. The record displays nicely the semantics of verb-concatenations, i.e. the list of PSMODS (nce moog .) contains the modifying verbs of the HEAD (dlha), i.e. the main verb.
Furthermore, the sentential record shows that additional functional information, such as 'SUBJECT', gets added to the record during processing. At this level of processing, any records at any depth could be displayed.

Error Correction

Ungrammatical sequences, such as DIR + DIR or DEICT + DIR, lead to a fitted parse, unless the blocking condition in the rules get relaxed/ ignored.
What exactly do we mean by "relaxation of the rules"? How does error correction work? In order to be able to correct deviant input, we need to foresee and implement expected errors. An international problem at this point is the lack of valuable error analyses or statistics for errors at the sentence level.
Regarding a syntactic restriction common to many languages - that of subject-verb-agreement -, we will demonstrate how this restriction is implemented in the grammar and how error detection works.

According to the PLNLP convention this condition is expressed as follows:

```
NP(conditions,...)
   VP(conditions,...,
      <PERSNUMB(NP).AGREE.PERSNUMB!SETERR<'PERS1'>>)
   -->
    VP(...,SUBJECT=NP,...)
```

Fig.20: Abbreviated sample PLNLP rule

The condition in angular brackets is to be read: "The person and number attributes of the NP have to agree with the person and number attributes of the VP OR (exclamation mark) the SETERR-attribute (the error correction mechanism) is set to a predefined error-type (here: 'PERS1'). This means that the conditions (first part of the angular brakket) are either met during the parse and the rules can be applied up to the final sentence level, OR the conditions will be relaxed during a second parse of the input string, during which the error correction mechanism is turned on and the second part of the OR-condition is invoked. Thus the grammar signals certain types of errors and triggers "repair activities" to be performed by the encoding (= generation) rules - or at least pinpoints the error.[10]

A syntactic error that is blocked by the rules for Hmong Njua is the incorrect verb ordering in a verb-concatenation. The error detection works similar to the agreement-error described above. The following rule and condition blocks both DIR + DIR and DEICT + DIR:

```
VP  VP(DIR,<MOTION(VP#1)!SETERR<'CONCAT1'>>)  -->  VP
```

Fig.21: Condition blocking incorrect verb-ordering in verb-concatenation

as it will be read: "Conjoin 2 VPs, the second of which is of type DIR (directional verb). Furthermore (the comma is an AND-condition), the first VP (VP#1) has to be of type MOTION, OR trigger a certain error message and repair activity (of type 'CONCAT1')", i.e. if the first VP is of type DIR or DEICT, the error is detected and a special error correction is triggered.

---

[10] For more examples see: Harriehausen (1989 b). For a detailed description of the entire system and its application (CRITIQUE) see: Richardson/Braden-Harder (1988).

Let us now look at an example that will lead to a fitted parse unless the error switch is turned on:[11]

```
 Enter a sentence or a PLNLP command:
moog nce.
---------------------------------------
XXXX1  VP1*    VERB1*  "moog" (go - DEICT)
       VERB2   "nce" (climb - DIR)
       PUNC1   "."
---------------------------------------

 Enter a sentence or a PLNLP command:
nwg moog nce.
---------------------------------------
XXXX1  VP1*    PRON1   "nwg" (he/she/it)
               VERB1*  "moog" (go - DEICT)
       VERB2   "nce" (climb - DIR)
       PUNC1   "."
---------------------------------------
```

Fig.22: Fitted parses of incorrect verb-ordering (DEICT+DIR) in isolation and embedded in a

Now let us relax the conditions and activate the error switch:

[11] One might think that the following example should be parsed with the main verb meaning of "moog" ("go"), i.e. as a verb of MOTION, followed by "nce", a verb of DIRection. Nevertheless, this example is incorrect, as native speakers would choose a different verb from "moog" to express the kind of motion in this case.

```
 Enter a sentence or a PLNLP command:
(setq nlperrs 1)        <-----  the flag gets activated
Value = 1

 Enter a sentence or a PLNLP command:
nwg moog nce.
-----------------------------------------
DECL1  PRON1   "nwg" (he/she/it)
       VERB1*  "moog" (go - DEICT)
       VP1     VERB2*  "nce" (climb -DIR)
       PUNC1   "."

GRAMMATICAL ERROR IN: SENTENCE 3.
  WRONG VERB-ORDERING IN CONCATENATION.
    nwg moog nce .
  CONSIDER:
    nwg nce moog .
  WRONG ORDERING OF VERBS
-----------------------------------------
```

Fig. 23: Error-correction of incorrect verb-ordering in concatenated structure

The incorrect syntactic structure gets parsed and leads to a complete parse. Apart from giving a description of the error ("Wrong ordering of verbs"), the problematic part of the input string is highlighted and corrected.

It should be emphasized at this point that the PLNLP grammars do NOT describe structures with non-expected errors. That is, for this type of input the rules cannot be applied up to sentence level, and by turning the error switch on, the deviation(s) cannot be cleared away. However, the system does not despair, because the FITREE procedure for fitted parses takes advantage of the fact that a lot of "intermediate junk" is produced during the parsing process. It uses the biggest pieces to come up with at least something instead of, for instance, lapsing into an infinite loop or returning a plain "FAIL".

Benefits

Apart from the advantages mentioned earlier for the fitted parse, the facility of assigning fitted parses and (optionally) correcting errors is extremely useful for several additional reasons. Let us summarize various reasons at this point:

- It allows for syntactic processing to proceed in the absense of a perfect parse.
- It provides a promising structure for further processing.

- The fitted parse diagram can be used by the grammar writer as an aid to rule-debugging.
- The flexibility of the approach enables the grammar writer to check for errors within the smallest and largest imaginable constituents.
- The correction device can be used for teaching second language acquisition.
- The user can use the system to improve his or her knowledge of the grammar of the language (imagine applications not only in second language acquisition, but also in high school or university courses for native speakers).
- It can be used for stylistic analyses.
- It can be used to enhance text processing systems with an error-correction device beyond the word level.

Acknowledgements

I would especially like to thank Karen Jensen, George Heidorn, and certainly many others in the PLNLP Group for their inspiring ideas.

References

Harriehausen, B. 1989 a. Hmong Njua. Syntaktische Analyse einer gesprochenen Sprache mithilfe datenverarbeitungstechnischer Mittel und sprachvergleichende Beschreibung des südostasiatischen Sprachraumes. Niemeyer Verlag.

Harriehausen, B. 1989 b. "Don't give up on me OR Why grammars need to expand their scope of parsable input", Proceedings of Conference on Arabic Computational Linguistics, Kuwait, December 1989.

Heidorn, G.E. 1972. "Natural Language Inputs to a Simulation Programming System". Ph.D. dissertation, Yale University.

Li, Ch.N./S. Thompson. 1981. Mandarin Chinese. A Functional Reference Grammar. Berkeley: Univ. of California Press.

Matisoff, J.A. 1973. The Grammar of Lahu. Berkeley: Univ. of California Publications.

Richardson, S.D. and L.C. Braden-Harder. 1988. "The experience of developing a large-scale natural language text-processing system: CRITIQUE". In Proceedings of the Second Conference on Applied Natural Language Processing, Austin, TX, Feb. 1988.