Prospects for Computer-Assisted Dialect Adaptation

David J. Weber

Instituto Linguistico de Verano
Casilla 2492
Lima, 100 PERU

William C. Mann

Information Sciences Institute
University of Southern California
Marina del Rey, California 90291

This paper describes a project which has explored the feasibility of using a computer to perform a significant portion of the changes required to adapt text from one dialect to several others. This ongoing experiment has examined adaptation between various dialects of Quechua, finding that a computer program may be an important tool for adaptation. An experimental computer program was written and applied to text, and its output was field tested in five target dialects. Preliminary results indicate that preprocessing text with a computer may 1) enable informants who are not bi-dialectical (in the source and target dialects) to produce adequate adaptations without much coaching from the linguist/translator; 2) improve the quality of the resulting text; and 3) reduce time and effort—both in adaptation and in manuscript preparation.

1. Introduction

This paper describes experiments in computer-assisted adaptation of text from one dialect to several others. The principal purposes of the initial explorations reported here were:

1. To discover whether a computer program could convert text in a dialect unintelligible to an informant into easily read (possibly errorful) text in the informant's own dialect.
2. To discover what kinds of dialect difference information are needed to support an effective dialect-adapting computer program.
3. To discover classes of dialect changes not accounted for in a particular first-draft computer program, thereby to provide data for a detailed examination of whether each class of changes is suitable for performance by a computer program.

In pursuit of these goals, an experimental computer program was written and applied to text, and its output was field tested.

In this paper the nature of the language situation is discussed first, followed by a description of the computer program. Then, procedures for checking the computer-adapted text are described, followed by a discussion of the results of this checking. Finally, conclusions are stated.

2. The Nature of the Language Situation

The practical difficulty of dialect adaptation is primarily determined by the language situation.

2.1 The General Nature of the Language(s)

This experiment was carried out in the subgroup of Quechua called "central" Quechua by P. Landerman [1]. These languages/dialects have the following characteristics:

1. More of the structure of the language is in the morphology than in the syntax.
2. Much of the discourse structure involves the manipulation of the so-called "topic" marker and the "evidential" suffixes (the reportative, the assertative, ...).

0362-613X/81/030165-13$01.00

3. The category of each morphological unit strongly constrains what suffix may immediately follow that unit. For example, the intransitive verb root *aywa-* may not be followed by the case marker *-man*, which only follows nouns.
4. A significant amount of morphophonemics is involved in getting from sequences of morphemes to the correct sequence of phonemes. The two such processes which are most widely applied are first, *morphophonemic lowering*, whereby a high vowel (i.e., either /i/ or /u/) of certain suffixes becomes /a/ when one of a certain small class of suffixes follows and second, *foreshortening*, whereby certain suffixes, the final vowel of which is otherwise long, appear with short, final vowels when followed directly by certain other suffixes.

These characteristics are mentioned because the design of the program responds to them.

2.2 The Nature of the Dialect Differences

Six dialects were involved in this experiment. The *source dialect* (abbreviated SD), i.e., the dialect from which text was adapted, was Huallaga (Huánuco) Quechua (abbreviated HgQ).

The *target dialects* (abbreviated TD) are the following:

Panao (Huánuco) Quechua (PaQ)
Dos-de-Mayo (Huánuco) Quechua (DoQ)
Llata (Huánuco) Quechua (LlQ)
Yanahuanca (Pasco) Quechua (YaQ)
Junín (Junín) Quechua (JuQ)

The differences between these dialects are not trivial, as should become clear in the discussion below. Each of these would require separate reading materials, but it is expected that the material of one dialect of Huánuco could be adapted to the other dialects of that department.

To what extent these dialects are mutually intelligible (pair-wise) is an open question. To our knowledge, no formal tests have been made. The first author has asked several HgQ speakers if they can understand the PaQ speakers. Answers indicate that communication is difficult. That HgQ speakers and JuQ speakers cannot communicate effectively has been observed firsthand. A rough metric of difference is this: adaptation between these dialects has required roughly 800 changes per 1000 words. Whatever the case, little or no significance is attached to whether the dialects involved are mutually intelligible or not; it is sufficient that separate materials need to be prepared for the dialects, as established by an overall consideration of the language.

The dialect differences will now be discussed in greater detail under three headings: the phonological differences, the lexical differences, and the grammatical differences.

2.3 Phonological Differences

Table 1 indicates various *regular sound changes* (RSC's) which have affected these dialects.

There are also some sound changes which are not so regular. For example, the change by which a Vowel-Semivowel-Vowel sequence becomes a long vowel (e.g. compare HgQ *chaya-* with DoQ *cha:-* 'arrive' and HgQ *tiya-* with DoQ *ta:-*) is not entirely regular. Some changes affect very few words (for example, the change by which /s/ is lost intervocalically, e.g., HgQ *wasi* 'house' but DoQ *wayi*, HgQ *usa* 'flee' but DoQ *uwa*), and these are handled in the program as though they were simply lexical differences.

2.4 Lexical Differences

The SD and TD may use very different roots to express the same concept. For example, to express 'to recover (from an illness), to get well', in HgQ one uses *allchaka:-*, in LlQ one uses *aliya:-*, and in JuQ one uses *kachaka:-*; 'to gather' is expressed by *shunta-* in HgQ and by *qori-* in LlQ; the pre-adjective meaning 'very' (used as in 'very big') is expressed by *sumaq* in HgQ and by *sellama* in LlQ; 'to lie (tell a falsehood)' is expressed by *llulla-* in HgQ and by *kaski-* in YaQ and JuQ ... et cetera. Sometimes the SD and TD possess the same root, but its meanings differ. *Aru-* means 'to work' in HgQ but 'to cook' in dialects to the west (Huaras, Ancash); *yacha-* means 'to know (how to)' in HgQ, whereas the cognate in JuQ *(yatra-)* means 'to reside at'.

2.5 Grammatical Differences

The grammatical differences between dialects can be divided into two general types, the morphological differences (i.e. those which occur within a single word) and the syntactic differences (i.e. those which involve more than one word). Only the morphological differences will be discussed in this section. They are of several sorts:

First, in the central Quechua dialects, one of the most drastic morphological differences involves how the plurality of the subject or object of a clause is marked in the verb of that clause. In LlQ there is only one verbal plural marker, *-ya:*. In HgQ there are five suffixes which indicate plurality within the verb. There are conditions on the occurrence of these, e.g., *-rka* occurs only preceding the continuative suffix *-yka:*. LlQ has only *-ya:*, HgQ has all except *-ya:*, and the other dialects have only *-:ri, -pa:ku,* and *-rka*.

	*č>ć (ch>ts)	*ç>č (tr>ch)	*ć>s (ts>s)	*λ>l (ll>l)	*ñ>n (ñ>n)	*q>h (q>h)	Phonemic Change / Orthographic Form
PaQ	−	−	−	−	−	+	
HgQ	−	+	−	−	−	−	
DoQ	+	+	−	+	+	−	
LlQ	+	+	+	+	+	−	
YaQ	+	−	−	+	+	−	
JuQ	−	−	−	+	+	−	

Table 1. Regular sound changes.

Second, there are differences in the properties of morphemes. For example, in HgQ the suffix *-:ri* undergoes morphophonemic lowering, but in JuQ it does not. In some dialects the durative suffix *-ra:* foreshortens while in others it does not.

Third, a suffix in one dialect may simply be absent in another. For example, LlQ has a verbal suffix *-ski*, a suffix which HgQ not only lacks, but for which there does not seem to be any corresponding suffix. HgQ has a suffix *-paq* which is used with future verbs, while YaQ has no such suffix.

Fourth, a single suffix in one dialect may correspond to two different suffixes in another dialect. For example, the relativizer *-sha* of HgQ corresponds to both *-sha* and *-nqa* in JuQ, which has a temporal contrast in the formation of relative clauses which is not present in HgQ. A similar case arises in the tense systems of these two dialects: JuQ has a distinction between a recent and a remote past tense which HgQ lacks.

Fifth, a suffix in one dialect may be the collapse of two suffixes in another. For example, in some dialects, the suffix *-mi* followed by the postposition *ari* has formed a single suffix *-mari*.

Sixth, suffixes may occur in different orders in different dialects. For example, in HgQ *-rqu* precedes the object marker *-ma:* (e.g. *maqa-rqu-ma:-nki* 'you hit me (recent past)'; in Huaras (Ancash) Quechua, it follows the object marker: *maqa-ma:-rqu-nki*.

Seventh, the number of allomorphs of a suffix may differ. For example, in HgQ the second person possessive suffix has allomorphs *-yki, -ki,* and *-niki*. JuQ has all of these forms as well as *-y*.

Finally, a suffix may have different forms (spellings) in the different dialects. For example, the suffix which expresses similarity is *-naw* in HgQ, *-noq* in LlQ and DoQ, and *-nuy* in JuQ; the continuative suffix is *-yka:* in HgQ and *-ya:* in JuQ; the first person plural conditional form is *-shwan* in HgQ and *-chwan* in JuQ.

Not all of the morphological differences between dialects have been mentioned in this section. The computer program handles all of the kinds mentioned, as well as others.

3. The Nature of the Computer Program

An early pencil-and-paper experiment in adapting from HgQ to the Quechua of Jacas Grande showed that, of a set of changes suggested by certain Jacas Grande speakers, approximately 95 percent were phonological, lexical or morphological, and a significant proportion of the residue were idiosyncratic rather than systematic. This suggested that making the program responsive to features beyond word boundaries would not make the task of final correction of an adapted text significantly easier. Thus the decision was made to have the program treat one word at a time.

The program is designed for Quechua, but not for any particular dialects. In all of the experiments to date, the source dialect has been HgQ.[1]

The program has three distinct phases of operation: initialization, derivation of target-dialect root spellings, and text-processing, each to be discussed in the following sections.

3.1 Initialization

In initialization, linguistic information about the dialects involved is made available to the computer program. We will discuss the information provided for a source dialect, and then what is provided for a target dialect.

[1] After this paper was written in 1979, more experiments with these dialects were performed. Experiments were also carried out in other dialects, including some Bolivian dialects of Quechua which were very different from these. There was also an independent series of experiments with Mayan dialects of the Quichean group in Guatemala. The program described here was modified to accommodate the other Quechua dialects. A new program was written for the Mayan dialects, using the same general design. All of these experiments met the same general goals as the experiments described here. This paper is an abridgment of the 1979 version [3].

3.1.1 Source Dialect Initialization

For the SD, two major lists and several small lists are provided to the program. The largest list is a dictionary of roots.

For each root, the dictionary entry (DE) contains the following information:

1. The *spelling* of the root, i.e. the string of characters by which the root is recognized.
2. The *name* of the root:
   a. If the root is native to Quechua, the name is the proto-Quechua form of that root.
   b. If the root is a Spanish loan, the name is the Spanish (phonemic) form of that root.
3. An indication of whether the root is the result of RSC's applied to the proto-Quechua form (negative for Spanish loans, names, ideophones, etc.).
4. Morphological category (e.g. noun, intransitive verb, transitive verb, etc.).
5. An indication of whether its final vowel is short or long.
6. An indication of whether its final vowel has undergone the morphophonemic process of lowering (from /i/ or /u/ to /a/).

Note that the dictionary will contain multiple DEs for the various allomorphs of the same root (i.e., for the various forms that the root may take). For example, a DE corresponding to one of the allomorphs of *yayku-* 'to enter' has (1) *yayku* as the spelling, (2) **yayku* as the proto-Quechua form; (3) it undergoes the RSC's; (4) it is an intransitive verb; (5) its final vowel is short; (6) it has not undergone morphophonemic lowering. All of this information is exploited by the program as described below.

The second major list is the list of SD suffixes. Each suffix DE contains the following information:

1. The spelling of the suffix.
2. An arbitrary name for the suffix, this name to be common to all the DE's corresponding to various allomorphs of that suffix and used uniformly for both SD and TD suffixes.
3. The morphological category of the unit (root followed by zero or more suffixes) to which this suffix can be applied.
4. The morphological category which results from concatenating this suffix to an appropriate unit.
5. An indication of whether its final vowel is short or long.
6. An indication of whether it disallows an immediately preceding long vowel.
7. An indication of whether its final vowel has undergone the morphophonemic process of lowering.
8. An indication of whether it causes morphophonemic lowering.
9. A number which gives the "order class" of the suffix, used to constrain sequences of suffixes to occur in a monotonically non-decreasing order of their order class numbers.

For example, the suffix *-ka:* (1) with arbitrary name /passive/ (2) applies to transitive verbs (3) with the result that the combination is an intransitive verb (4); the final vowel is long (5) and the suffix does not disallow a preceding long vowel (6); the final vowel has not undergone morphophonemic lowering (7) and it does not cause such lowering (8); the order class is 600, placing it in the large class of "derivational" suffixes (not ordered with respect to other members of the class, but ordered with respect to other "order classes" of suffix).

In addition to these two major lists, there are several small lists:

1. The allomorphs which can only occur in word-final position following a short vowel; one such DE has as its spelling *m* and arbitrary name /MI/; there is another DE in the suffix dictionary which has *mi* as its spelling and the same arbitrary name. These two DE's correspond to the two allomorphs of the suffix *-mi* 'assertive'.
2. The categories of morphological units which are allowable in word-final position, i.e. the categories of complete words.
3. The table of orthographic changes, used to convert the SD expression of vowel length to a single form, and to remove SD orthographic reflections of low level phonetic processes.

Source dialect initialization is independent of target dialect information, and so is invariant with change of target dialect.

3.1.2 Target Dialect Initialization

For a given TD, a dictionary of roots, a dictionary of suffixes, and several other small lists, must be provided the program.

The small lists which must be made available are:

1. The *non-cognate roots list*: corresponding SD and TD roots where the SD root and

the TD root are not cognate. For example, the HgQ root *lloqshi-* 'to leave' corresponds to the YaQ root *yarqu-*.

2. The *plural suffix list*: all verbal suffixes in the SD which indicate that the verb so marked has a plural subject or object.
3. The *drop list*: suffixes to be deleted in adapting from the SD to the TD. For example, HgQ *-paq* is absent in YaQ and must be dropped in adapting from HgQ to YaQ, since there is nothing in YaQ to which *-paq* corresponds. All of the SD pluralizers must be on this list.

3.2 Target Dialect Root Derivation

TD roots are derived from SD roots so that it is not necessary to enter a dictionary for the target dialect. This eliminates the arduous task of collecting, organizing, and entering such a dictionary, and second allows one to begin adaptation on the assumption that the target dialect roots are related in some systematic way to the source dialect roots.

TD roots are derived from the SD roots as follows: The DE of each SD root includes the proto-Quechua form of that root; the RSC's are applied to this form. For example, suppose that the SD root is *chaki* 'dry'. In the DE for that root the proto-Quechua form is given as **čaki*. Now if the TD is DoQ, the change č>ȼ from the table of RSC's for that dialect is applied to the proto-Quechua form; the result (after orthographic adjustment) is *tsaki*. If, however, the TD were LlQ, the changes č>ȼ followed by ȼ>s would be applied to the root with the resulting LlQ word *saki*. (These changes could be made as one change, č>s, but it is no less convenient to handle it as two steps, which is clearly how the change came about historically). The process uses a simple substring substitution algorithm.

After the RSC's have applied, a TD DE is made for that root by substituting the derived spelling for the SD spelling.

Three types of roots must not undergo the RSC's: (1) roots for which the TD root is simply not cognate with the SD root, (2) native Quechua words which do not undergo the sound changes, such as names (personal, place, . . .) and ideophones, and (3) words borrowed from Spanish.

3.3 Text Processing

After initialization, a TD text may be produced from a SD text. The text is treated word by word. Each word passes through 1) an analysis process, and 2) a synthesis process, which are independent.

3.3.1 Word Analysis

Word Analysis (WA) transforms a word to a set of distinct *readings*, each reading consisting of a sequence of morpheme names and an indicator marking the presence or absence of pluralization.

The first step in WA is rewriting vowel length (which in the practical orthography is written with double vowels, e.g. "aa"), as a vowel followed by a colon. This is necessary because the vowel and its length may not be parts of the same morpheme. For example, *aywaa* 'I go' is composed of the morphemes *aywa-* 'to go' and *-:* 'first person'. The change makes it possible to recognize all morphemes according to their spellings.

WA seeks to decompose a word into a sequence of morphemes. It works from left to right, first seeking a root whose spelling matches the beginning of the word, and if successful, then seeking a suffix whose spelling matches some immediately following sequence of characters, repeating this process until all the characters of the word have been matched by the spellings of DE's. WA is organized in such a way that it considers all possible sequences of DE's (root followed by any number of suffixes) whose spellings—taken together in sequence—match the characters of the word. The algorithm is a computationally-straightforward recursive exhaustive search.

However, this character-matching process is not a sufficient basis for arriving at correct analyses; many sequences of morphemes whose spellings (taken together) match the characters of the word will be unacceptable, because the language imposes constraints on the allowable decompositions. To eliminate spurious analyses, the program's ongoing analysis is correspondingly constrained. For example, if character matching alone is used to analyze *aywaykan*, there are 56 decompositions into morphemes. If, however, the constraints are imposed, then only one of these decompositions is passed as legitimate.

The constraints incorporated into WA are of two types. The first type is invoked whenever there is a match with some suffix; such constraints test the acceptability of that suffix as the successor of the immediately preceding morpheme (be it root or suffix). For example, one possible decomposition of *chakiykan* 'it is drying' is the root *chaki* 'foot' (from the proto-Quechua root */ɕaki/*) followed by *-yka* 'continuative' and then by *-n* 'third person'. At the stage where the suffix *-yka* is matched, the constraints include a category check, which fails in this case because the root is a noun and the continuative suffix may only follow verbs. By contrast, when the match is to the root *chaki* 'to dry' (from the proto-Quechua root */čaki/*) and then *-yka* is matched, this same test does not fail and this decomposition goes on to be the successful

analysis. Such tests help reject bad decompositions before a great deal of computation is spent on them.

The second type of constraint is invoked when the word is completely decomposed into morphemes. For example, the word *aywaykaran* 'he was going' can be decomposed as *aywa-yka-ra-n* where *-yka* is taken to be one of the allomorphs of the suffix /-YKU/. This decomposition passes all of the constraints which apply to adjacent morphemes, but not the overall constraint that certain allomorphs (*-yka* as an allomorph of /-YKU/ among them) are appropriate only if one of a certain small class of suffixes (*-chi, -mu,* etc). follows somewhere in that word. The correct analysis involves the same morpheme boundaries but with *-yka* as an allomorph of the suffix /-YKA:/ 'continuative': *aywa-yka-ra-n* (go-continuative-past-third:person).

All morpheme decompositions of a word which pass the tests are collected. Any morpheme sequence containing a plural morpheme is marked as plural, and the plural morpheme is deleted. The resulting set of marked, depluralized readings is the final result of WA.

When WA does not yield any reading, the SD word passes unchanged through the remainder of the program and appears in brackets in the final text. If the word fails to be analysed because its initial characters do not match any root in the dictionary, that word is entered on a list which is periodically reviewed to upgrade the dictionary of roots.

3.3.2 Word Synthesis

Word Synthesis (WS) derives a TD word for each reading produced by WA. What passes from WA to WS for each reading is a sequence of morpheme names and the plurality tag of each sequence. From this sequence are dropped any morpheme names which are listed in the drop list, and substitutions of non-cognate TD roots are made. For the remaining morphemes, each name is replaced by its entire TD DE (so that its properties are characteristic of the TD rather than the SD). The next step is repluralization.

Repluralization inserts a plural morpheme into the morpheme sequence according to the pluralization scheme of the TD. Repluralization is dialect specific, and requires different methods for different dialects. The method appeals to the order of various classes of suffixes to find the appropriate niche for the pluralizer.

The next step is to select for each morpheme the allomorph which is compatible with the other morphemes in the word. The allomorphs of a morpheme are represented by DE's which share the name of the morpheme. This set of DE's is reduced by successive application of constraints until a single DE remains. When the choices have been made for all the morphemes of the sequence, the spellings of the chosen DE's are concatenated to form the TD word.

Then all of the separately developed renderings of a word are collected, and any duplicates are eliminated. Ordinarily, this results in a single word. If there are multiple renderings, they are separated by slashes(//). For example, HgQ *kasha* has two possible analyses: 1) the root *kasha* ('thorn') and 2) the root *ka* ('be') followed by the suffix *-sha*, (past participle). In going to LlQ, the first of these analyses will yield *kasha*; the second will yield *kashqa*. Thus the resulting LlQ text will have *kasha//kashqa* corresponding to HgQ *kasha*.

The final step is to convert the word(s) to the TD orthography, representing length by doubled vowels and making any other necessary changes.

3.4 The Program's Effects on Single Words

This section roughly characterizes how the program treats certain actual words, illustrating how certain dialect differences are handled. The forms given below are only suggestive of how the program operates; they are not the actual form of the word in the computer, and possible alternative readings are not shown. (Capital letters are used for the arbitrary names of suffixes; the asterisk preceding roots indicates that the form is a proto-Quechua form). The pair of examples in Table 2 illustrates the rather dramatic difference which arises in words containing pluralizers in adapting from HgQ to LlQ.

In (1), *-rka* is a pluralizer, whereas in (1') it is the allomorph of -RKU which has undergone morphophonemic lowering, in this case because it is followed by -:RI. In the first example *-rka* is eventually dropped because it is a pluralizer; in the second it eventually becomes *-rku* because once -:RI is dropped there is no longer any reason for the vowel to be lowered. Note that in the second example, the pluralizer which is inserted (-YA:) occupies the same position as the SD pluralizer (-:RI), but in the first example the inserted pluralizer occupies a different position than the SD pluralizer (in this case, *-rka*). Further note that the allomorph selected for the pluralizer -YA: has length in the second example but not in the first; this is because in the second example the following suffix (-NA) does not foreshorten, whereas in the first example the following suffix (-3) does. A similar case involves -YKA: in the first example: in the SD the correct allomorph is *-yka* because it immediately precedes *-n*, which foreshortens, whereas in the TD the correct allomorph is *-yka:* because it immediately precedes -YA:, which does not foreshorten.

Four tables of examples will now be given (with fewer stages) which illustrate cases in which a TD root replaces an SD root which has different properties

Given:		
SD orthographic form:	(1) aywarkaykan	(1') aywarkaarinanpaq
WA develops, in succession:		
length converted:	(2) aywarkaykan	(2') aywarka:rinanpaq
segmentation:	(3) aywa-rka-yka-n	(3') aywa-rka-:ri-na-n-paq
morphophonemic form:	(4) *aywa-RKA-YKA:-3	(4') *aywa-RKU-:RI-NA-3P-PAQ
plurality handled:	(5) (*aywa-YKA:-3)+PL	(5') (*aywa-RKU-NA-3P-PAQ)+PL
WS develops, in succession:		
re-pluralization:	(6) *aywa-YKA:-YA:-3	(6') *aywa-RKU-YA:-NA-3P-PAQ
allomorph selection:	(7) aywa-yka:-ya-n	(7') aywa-rku-ya:-na-n-paq
TD orthographic form:	(8) aywaykaayan	(8') aywarkuyaananpaq
	'they are going'	'in order that they go'

Table 2. The program's effects on two plural words.

SD orthographic form:	warantin
segmentation:	wara-ntin
WA output:	(*wara-NTIN)-notPL
re-pluralization:	*wara-NTIN
allomorph selection:	waray-nintin
TD orthographic form:	waraynintin
	'day after tomorrow' (< *wara(y)* 'tomorrow')

Table 3. Consequences of root change.

SD orthographic form:	chayaykun	chayamun
segmentation:	chaya-yku-n	chaya-mu-n
morphophonemic form:	(*ça̧ya-YKU-3)-notPL	(*ça̧ya-MU-3)-notPL
re-pluralization:	*ça̧ya-YKU-3	*ça̧ya-MU-3
allomorph selection:	ça̧-yku-n	ça̧:-mu-n
TD orthographic form:	traykun	traamun
	'he arrives (directly)'	'he arrives (here)'

Table 4. Selection of root allomorphs for length.

than the TD root. In the example of Table 3 the SD root ends with a vowel, whereas the TD root ends in a consonant. The suffix *-ntin* has an allomorph *-nintin* which occurs following consonants and long vowels, so this is the appropriate allomorph for the TD.

In the examples of Table 4 the SD root *(chaya-)* has but one allomorph, but in the TD (YaQ) the corresponding root has both long and short allomorphs. In the first example of this table the long form does not appear because the root immediately precedes *-yka*, which foreshortens. In the second example the length occurs because *-mu* does not foreshorten.

In the examples of Table 5 the SD root *(lloqshi-)* is replaced in the TD (YaQ) by a root which has two allomorphs. In the first example here the final vowel of that root is /a/ because of morphophonemic lowering caused by *-mu*; in the second it is /u/ because no

SD orthographic form:	lloqshimun	lloqshikun
segmentation:	lloqshi-mu-n	lloqshi-ku-n
WA output:	(*lloqshi-MU-3)-notPL	(*lloqshi-KU-3)-notPL
re-pluralization:	*yarqu-MU-3	*yarqu-KU-3
allomorph selection:	yarqa-mu-n	yarqu-ku-n
TD orthographic form:	yarqamun	yarqukun
	'he leaves	'he leaves
	(said from within)'	(said from without)'

Table 5. Selection of root allomorphs for lowering.

SD orthographic form:	suwamannaw	
segmentation:	suwa-man-naw	suwa-ma-n-naw
WA output:	(*suwa-MAN-NAW)-notPL	(*suwa-MA:-3-NAW)-notPL
re-pluralization:	*suwa-MAN-NAW	*suwa-MA:-3-NAW
allomorph selection:	suwa-man-noq	suwa-ma-n-noq
TD orthographic form:	suwamannoq	

Table 6. A single word form resulting from an ambiguous analysis.

suffix follows in the word which causes morphophonemic lowering.

In the example of Table 6 the SD word is ambiguous (between 'as though to a thief' and 'as though to steal me'). It is seen that this ambiguity produces only one TD word, a word which is ambiguous in the TD in the same way as it is ambiguous in the SD.

3.5 Program Design Problems: Ambiguity and the Control of Complexity

The problem of ambiguity strongly influenced the design and the allocation of development effort. We now feel that ambiguity control is one of the major considerations in designing an adaptor. The work of coping with ambiguity and complexity is presented below in a series of retrospective (non-chronological) design decisions.

Given our previous decision that the program should work on a word-by-word basis, one could (in principle only) design an adaptor which would work perfectly, without any internal linguistic knowledge except a table of word correspondences. The program would, for each given word, look up the adapted word. The difficulty with this for Quechua is the practical impossibility of providing the word table. Morphological productivity would yield a table of millions of words, simple in structure but of unmanageable size. Also, the process of building the word table would continue indefinitely, and much of the relevant, systematic knowledge could not be used at all.

This perception leads to the initial design decision:

> *Decision 1:* The basic linguistic unit which the program manipulates should be the morpheme rather than the word.

Since we would like to be able to adapt from many SD's to many TD's with minimal effort of change, this leads to another decision on the particular morphemes to be used:

> *Decision 2:* The program should analyze text into proto-Quechua morphemes where appropriate, and should synthesize text from proto-Quechua morphemes by recapitulating actual historical processes.

Discovering words in text is trivial, but discovering distinct morphemes is not. The decision to use morphemes as the basic unit makes it necessary for the program to contain an analyzer to reduce individual words into sequences of morphemes. This is the major part of the Word Analyzer above.

One might hope that distinct morphemes would have distinct spellings, and that they could be discovered effectively by seeking concatenations of morpheme spellings which would yield the word. Unfortunately, as we have seen, spelling is a hopelessly inadequate basis for morphological analysis. Use of spelling alone leads to truly amazing numbers of analyses, as was demonstrated above with *aywaykan*. This is so frequent that, for efficiency, the program must never even produce the set of spelling-based readings inter-

nally. Rather, the analysis must be constrained by knowledge other than spelling. This fact leads to another design decision:

> *Decision 3:* The program must incorporate a strongly constraining morphology in its analysis.

The morphology used [2] makes it possible to exploit the characteristic of Quechua (mentioned in 2.1) that the category of a morphological unit strongly constrains what suffix may follow the unit. On the basis of the morpheme categories alone, the number of distinct analyses of *aywaykan* drops from 56 to 2. The program applies these morphological constraints as the analysis of a word proceeds, so that *an invalid analysis can be rejected as soon as it is proposed* if it contains an invalid pair of adjacent morphemes, rather than delaying rejection until the word analysis is complete.

The program also uses several other kinds of criteria for rejecting analyses, including a set of programmed tests which apply to whole words, used as soon as an analysis is complete. They eliminate analyses containing defects such as incorrect lowering, improper ordering, and the improper use of certain special morphemes such as *-paq* 'future'.

These methods eliminate nearly all of the potential ambiguity (well over 99 percent of it), but still leave a significant residue. In the experiment, about 25 percent of the words were still ambiguous under all of these constraints applied jointly, and this ambiguity contributed about 1 excess reading for every 2 words. If this amount of ambiguity were to appear in the final text, it would be a significant impediment. Fortunately, as we will see, most of it goes away.

In synthesizing words in the target dialect from their morpheme definitions, the many allomorphs and associated conditions must be treated properly. A TD morpheme may have multiple allomorphs in surface forms of the TD, only one of which is allowed by conditions in the remainder of the word. There is a great diversity of conditions. Composing TD words without reference to these conditions would yield incomprehensible text. This leads to the final design decision:

> *Decision 4:* The program must contain linguistically sound methods for selecting correct allomorphs in context.

About one third of the words in our experimental text had possible allomorph variants when adapted to JuQ. Virtually all of this variation is subject to conditioning, which can be represented computationally by systematic environmental filtering rules. Application of this elaborate collection of environmental filtering rules causes essentially all allomorph variation to be resolved during synthesis.

The final ambiguity-resolution "method," the one which resolves about 93 percent of the residue, is not really a method at all. It turns out that when the various readings of an ambiguous SD word are synthesized in the TD, they seldom differ. The predominant case is that only one spelling (rendering) arises from an ambiguous word, so that the final text has only a single word form even though the analysis and synthesis of that word were ambiguous. On a 576-word sample of text adapted to JuQ, there were 151 ambiguous words. For these, WA produced 850 readings (1.47 readings per word). But the "collapsing effect" reduced the number of readings to 587 (1.02 readings per word), so that only 11 words were indicated as ambiguous in the final text.

In checking and correcting these texts, native speakers never had any difficulty in quickly selecting one of the alternative words as being correct, and agreed on all selections. The number of alternatives in the resulting TD text did not present difficulties for the checker/editors. The final level of ambiguity presented no experimental difficulties of any kind.

Given the necessary complexity and diversity of the symbol processing required by the design decisions above, the programming language and related software support must provide:

1. Flexible facilities for manipulating character strings,
2. Easy representation of processes with elaborate functional decomposition,
3. Recursive processes,
4. Strong tools for defining, testing, revising and combining processes.

These requirements (among others) led to our choosing INTERLISP (a dialect of LISP) as the programming language. LISP encourages functional decomposition of one's program; this has been vital to keeping the whole enterprise comprehensible and controllable. This paper's description of the program does not reflect the strong functional decomposition employed. There are actually 111 functions in the current version of the program, and a similar number of other (tool) functions were defined for managing dictionaries, printing statistics and other activities.

4. Procedures for Checking the Computer Adapted Text

All during the months-long development of the program there was an understanding that its effective-

ness would have to be examined in some sort of field test. Subsequent development and application would depend on the outcomes of the testing, i.e. on how readable and correctable the computer-produced text really is. In particular, we were prepared to embed the present program in a more complex program whose scope was sentences or larger units if that had proved necessary.

To test the program's adequacy, samples of various types of text were selected for trial adaptation, and all samples were adapted into five target dialects. In the summer of 1978 these texts were taken by the first author to the areas where the target dialects are spoken. They were presented to native speakers of the various dialects, and their reactions and suggestions for revision were noted. This section presents the testing, and Section 5 describes the results.

4.1 The Nature of the Texts Used in the Experiment

The source texts used for this experiment consisted in the following: two folktales "Hombre y Oso" (approximately 150 words) and "Moco y Mishi" (approximately 250 words), two passages from the translation of the *Gospel of Mark* 2:1-12 (The Healing of the Paralytic) and 14:12-15:41 (The Last Supper through The Death of Jesus), and a short personal narrative about an accident.

4.2 Procedure Used to Check the Texts

The procedures used to check these texts varied from helper to helper as a function of his ability to read and to grasp what was expected of him by way of editing or correcting. The most knowledgeable helper took pen in hand and proceeded to correct and edit with intervention (by the first author) only on certain matters of spelling (where the official orthography — which was not known to him — was in conflict with his intuitions based on the writing of Spanish). This helper fully grasped the task of editing: he would frequently back up (as he proceeded through the texts) to read several sentences before the one he was focusing on to see that the passage read smoothly. He made a fair number of changes in word order and some that were clearly intended to bring the discourse structure of the text more into line with his conception of what it should be. (This raised an interesting problem, one which has no obvious solution, namely how can one distinguish between those changes which are essential to the intelligibility and good style of the TD text from those which make the text conform to a personal style? And how would one train a checker/editor to distinguish between these?).

The most common situation was that the helper was illiterate (or that he had rather minimal reading skills in Spanish). In this case it was necessary to read the text to him sentence by sentence, pausing and soliciting corrections, suggestions, and impressions if these were not forthcoming, and monitoring his understanding of the text. The main drawback of this method is that it was difficult (usually nearly impossible) to get good reactions as to the discourse features of the text. (And on a second reading, the helpers were unwilling to criticize the text which they had just corrected).

Some text was checked merely by handing it to a person (who in each case was a teacher at the secondary or university level) with the instruction to correct it—to make it good text for his dialect. In these cases very little was done; far fewer changes were made than in any case where a linguist worked with the corrector/editor.

Despite these and other drawbacks, it was felt that, by and large, the checking and correcting did bring the TD texts extremely close to what would be adequate even for Scripture translation. (Of course, this is a subjective judgment).

5. Results of Checking the Computer Adapted Text

Checking the computer-adapted text with native speakers of these dialects has revealed both the general strength of the program an an interesting residue of disparities between computer-produced text and text after subsequent corrective editing.

5.1 General Results

The results of testing the computer-adapted texts in the areas where the various target dialects are spoken has been very encouraging.

First, thousands of changes were correctly made by the program. Samples of the text adapted to LlQ were counted to estimate the program's quantitative effects. It changed about 760 morphemes per 1000 words of text. The native helper suggested about 190 additional changes per 1000 words of text, and these changes include some which are editorial rather than dialect-related. (This was an extreme; all helpers for other dialects suggested far fewer changes of the same material). So the program seems capable of doing between 80 percent and 95 percent of the necessary or advisable dialect-related changes. These figures are of less practical importance than the qualitative fact that the program made these texts fully comprehensible to the native helpers.

Of the corrections which field testing brought to light, a rough estimate suggests that well over 90 percent could also have been made (and eventually will be made) by the program by simple modifications of the data lists made available to it. (The bulk of these changes involve the spelling of Spanish loans, which in the experimental tests were written as in Spanish but which will eventually be written according to Quechua orthography). This means that *only a very small residue of changes will not be made by the program,* these changes being left for the checker/editor.

The computer-adapted text was — with very few exceptions — completely intelligible to the checker/editors, and this would not have been the case for material which had not been pre-adapted.

Thus it seems safe to say that one consequence of computer adaptation is that the resultant text could be checked by a person who knew only the target dialect, his task being to make the text sound more natural in his dialect. On the same basis, computer adaptation can make an existing translation in a related language or dialect accessible to a native translator.

It has been our feeling that pre-adapting with a computer would enable the checker/editor to consider more significant changes because it relieved him of the (overwhelming number of) rather superficial and systematic changes. This experiment has not verified this (nor contradicted it) because those who helped were not trained in the task of checking/editing, which—it is assumed—will be the case in the eventual applications. (Training checker/editors, who will work from pre-adapted text, seems a reasonable task. By contrast, to train persons to make all the changes, working directly from the SD text, would be a formidable if not impossible task. But by first preadapting the text, that which must be done manually by the checker/editor, and consequently what he must be trained to do, is reduced to well within the feasible).

One implication of the results of this experiment is that the range of dialects for which there should be a single primary translation has been considerably enlarged. Mutual intelligibility is no longer an adequate basis on which to make decisions about the allocation of linguistic or translation personnel. Mutual intelligibility may be very low for some extremely trivial, superficial reason, and even if it is low because of a great many, seemingly complex factors, so long as they are systematic, computer adaptation may make otherwise unintelligible text quite intelligible to a speaker of the target dialect, intelligible enough that he can edit it without some other means of discovering its meaning.

5.2 Disparities Between Computer-Adapted and Final TD Texts

This section lists kinds of text changes which were discovered to be necessary to make the computer-adapted texts better conform to the TD's. One point seems inevitable: *there will always be a non-trivial residue of changes which cannot be done by the computer.* Thus, it will always be necessary that the TD texts be manually corrected or edited. So rather than strive to make the program do all the changing possible, it is wise to be content with a program which does a significant amount of the changes—those "low-level" changes which are systematic—and not unduly complicate the program by attempting to make it do an unrealistic amount. The proportion of all of the desired changes which is accomplished by the present program seems entirely satisfactory; no further development is needed to change the balance between manual and programmed changes.

It was gratifying to find that, although some of these residual kinds of changes (described below) were difficult or impossible to program, they caused little difficulty for the checkers in the field test. The raw computer-adapted text was as good as a conventional rough draft, and could have been used in the same way.

1. Changes due to cultural differences

The Quechua culture is not completely homogeneous throughout the Andes; there are cultural differences which give rise to certain problems in adapting from one dialect to another.

In HgQ, the expressions for the time of day are tied to the practice of chewing coca, e.g., *chaqcha inti* (coca:chewing sun) mid-morning coca break (roughly at 10:30 AM) and *mallway inti* or *mallway oora*, which refers to the time of the mid-afternoon coca break. In other areas—Junín, for example—where the chewing of coca plays a lesser part in the culture, time is not indicated by reference to coca chewing, presumably because there are no "institutionalized" times for chewing. In these areas time is told either by expressions borrowed from Spanish, e.g., *las tres* '3 o'clock' or by reference to the position of the sun, e.g., *inti yarpunan oora* (sun falling time) '5:00-6:00 PM'.

2. Different complements

In HgQ, the complement of a *phasal* verb[2] is an infinitive (object) complement; e.g., *miku-y-ta qallaykusha* (eat-INF-ACC he:began) 'He began to eat.' But in LlQ, the complement to a phasal verb is an adverbial clause, e.g., *miku-r qalaykusha* (eat-ADV he:began) 'He began to eat.'

3. Case

In HgQ forgiveness is expressed by the verb *perduna-* (borrowed from the Spanish *perdonar*) and that of which one is forgiven is expressed by an **accusative** noun phrase; e.g., *noqa qam-ta huchayki-ta perdunaami* (I you-ACC your:sin-ACC I:forgive:you) 'I forgive you your sin.' In DoQ, it is expressed by an **ablative** noun phrase: *noqa qam-ta huchayki-pita perdunaami* (I you your:sin-ABLATIVE I:forgive:you) 'I forgive you from your sin.'[3]

4. Spanish loans

The process of adaptation must deal with differences in what elements of Spanish have been borrowed and how they have been assimilated. So, adapting from HgQ, we have:

1. A Spanish loan changed to a native Quechua word, e.g., HgQ *alberti-* becoming LlQ *yaasi-* 'to advise'.
2. A Quechua word becoming a Spanish loan, e.g., HgQ *pachaku-* becoming LlQ *posadaku-* 'to take lodging'.
3. A Spanish loan becoming a different Spanish loan, e.g., HgQ *iwal* becoming PaQ *pareehu* 'together (with)', HgQ *sabli* (from Spanish *sable* 'saber') becoming YaQ *macheti* (from Spanish *machete*).
4. Spanish loans in different degrees of assimilation, e.g., HgQ *ligi-* (highly assimilated) becoming LlQ *liyi-* 'to read' from Spanish *leer* ; HgQ *rigi-* (highly assimilated) becoming DoQ *kriyi-* 'to believe' from Spanish *creer* ; HgQ *iskirbi-* becoming DoQ *iskribi-* 'to read' from Spanish *escribir* ; HgQ *pakillaa* becoming DoQ *Dyosolpaki* 'thank you' from Spanish *Dios se lo pague* 'May God repay you'.

5. Idioms

Idiomatic expressions vary from dialect to dialect (although very little). For example,

1. HgQ *nawinkuna fiyupa pununaptin* (their:eyes really because:they:desire:sleep) 'because their eyes really wanted to sleep' is an idiomatic way to say that they were extremely sleepy. This idiom is not used in LlQ, DoQ, and YaQ, where it is adequate merely to say *fiyupa pununaptin* 'because they were really sleepy'.
2. In HgQ, 'to greet' is expressed by *adyusta qo-*, literally, 'to give an adios'. In other dialects this is expressed simply by the Spanish loan *saluda-* 'to greet'.
3. In HgQ, to say that someone began to sob, one says *waqayman churakaran* (to:cry he:was:placed). In DoQ *waqayman chayarqan* (to:cry he:arrived).
4. The HgQ expression, *tapriypa tupaypa*, which means roughly 'end for end, head over heels' is unknown in e.g., JuQ.
5. In HgQ, roosters are said to sing *(kanta-)*, whereas in YaQ, to cry *(waqa-)*.

It was encouraging to note a case in which an idiomatic usage is preserved even though the verb root is changed. The case is HgQ *wasi-ta hata-ra-chi-* (house-ACC stand-RI-cause-) 'to raise a house' (literally, 'to cause a house to stand'). In the dialects to the east, the same idiom is used but with the local reflex of **shaya-* 'to stand'; thus in DoQ, 'to raise a house' is expressed *wayita sharkatsi-*.

6. Auxiliary constructions

HgQ has a construction which indicates that some action or event is imminent. Some dialects lack the imminent construction. For such dialects it was necessary to change this to the periphrastic future.

7. Suffixes present in one dialect but absent in another

In LlQ there is a fairly common verbal suffix *-ski* which HgQ does not have. In adapting from HgQ to LlQ it was left to the checker/editor to insert *-ski* where appropriate.

8. Pluralization

The pluralization algorithm works perfectly, in that whenever there is a pluralizer in the SD, the verb is appropriately pluralized in the TD. However, this treatment is deficient for the following reason: in HgQ (the SD for the present experiment) only about 30 percent (a rough guess) of those verbs which have a plural subject or object are actually marked as being plural in the verb. In other dialects a much greater percentage of the verbs which have plural subjects or objects are marked plural. For LlQ, this might be as high as 90 percent. In adapting from HgQ to LlQ, far too few pluralizers were inserted because there were too few pluralizers in the HgQ verbs to trigger the process which inserts them into the TD verbs.

[2] A phasal verb is one which refers to the starting, stopping, continuing, et cetera of the action expressed by its complement, e.g., *to start to eat, to stop talking,* etc.

[3] This difference may be due to HgQ having assimilated the loan more highly than DoQ; the ablative would be used in Spanish: *Yo te perdono de tus pecados* 'I forgive you from your sins.'

9. Differences in subcategorization

In one of the texts, the fox reports that the cat is "slaughtering" *(pishta-)* the lard jar. Virtually all the Quechua checkers from other dialects wished to change "slaughter" to "claw", this because only something animate can be slaughtered, or to put it in linguistic parlance, because *pishta-* 'slaughter' is subcategorized as having an animate object in those TD's, whereas it seems to be not so subcategorized in HgQ.[4]

10. Tense differences

In HgQ there is a simple past tense, marked by the suffix *-ra*; there is also an aspectual marker *-rqu*, which means roughly 'suddenly, momentarily'. In some dialects this aspectual marker seems to have become a recent past tense marker giving rise to a contrast between recent and remote past. This raises a problem for adapting from HgQ to such a dialect, namely, determining when it is appropriate to use the recent past and when the remote past.

5.3 Side Benefits of Using Computers in Adaptation

Many unexpected side benefits of computer aided adaptation have been recognized in the course of this experiment. First, *the process of checking adapted texts will identify many improvements that can be made in the source text.*

Second, the translations of a particular source text used in neighboring dialects will differ principally only in being dialect variants. More fundamental differences could lead to disputes about which is the more accurate or appropriate rendering of the text; dialect-based differences in texts are not likely to do so.

Third, the SD and TD texts will be more error-free than would otherwise be the case, in several ways. Because WA usually succeeds in analyzing a word only if it is spelled exactly right, the program has the effect of identifying essentially all of the typographical errors in the SD text. And since the computer does an impeccable job of fabricating the TD word, that word is spelled correctly, and consistently from instance to instance.

Fourth, the corrections required to convert the computer-adapted text into a final TD text are *not* so numerous as to require retyping of the text. This is not the case for most manual adaptations. The final text can be produced by on-line editing of the computer-produced text, with significant savings of proof-reading effort.

Fifth, the program leads to a greater understanding of 1) the structure of the SD, (since WA must contain a fully explicit morphology), and 2) the dialect situation. It provides a medium in which to express these insights in a fully explicit manner.

6. Conclusions

This project has explored the feasibility of using a computer to assist in the adaptation of material from one dialect to another, using central Quechua dialects as its test-ground. All the results suggest that a computer *can* be programmed to do a *significant* amount of adaptation, but this depends on incorporating a significant amount of knowledge about the source and target dialects into the program in order that it be able to analyze source dialect words, make changes, and reconstruct a correct target dialect word.

References[5]

1. Landerman, P., "The Proto-Quechua First Person Marker and the Classification of Quechua Dialects," 1978. Manuscript
2. Weber, D.J., *Suffix-as-Operator Analysis and the Grammar of Successive Encoding in Llacón (Huánuco) Quechua,* SIL, Peru 1976.
3. Weber, D.J., W.C. Mann, "Prospects for Computer-Assisted Dialect Adaption," *Notes on Linguistics* Special Issue No. 1, 1979. (Available from SIL, 7500 West Camp Wisdom Road, Dallas, Texas 75236, cost: $.75)

David J. Weber is a member of the Peru branch of the Summer Institute of Linguistics and is also a graduate student at the University of California at Los Angeles (UCLA). He received the M.A. degree in Linguistics from UCLA in 1978.

William C. Mann is a member of the research staff of Information Sciences Institute at the University of Southern California. He received the Ph.D. degree in computer science from Carnegie-Mellon University in 1973.

[4] This example is weak because probably *pishta-* is also so subcategorized in HgQ, the violation in this case being tolerable because it is somewhat metaphorical. Whatever the case, there will be some minor subcategorization differences from dialect to dialect.

[5] We have searched for references in the literature or other signs of comparable prior work on computer-aided dialect adaptation, and have found none.