Programming Aspects Of MT

Dan A. Belmore
Consultant Programmer
Institute of Languages and Linguistics

The general steps in MT research prior to programming might be
outlined thus:

Step 1: Linguistic analysis of the source and target languages.

Step 2: Expository description of behavior patterns of linguistic
components of the languages.

Step 3: Description by symbolic logic statements (Formulation).

For example:

Source sentence:	СТАКАН	ЦАРАПАЕТ	СТОЛ.
From glossary:	glass	scratch	table
From Russian "stem" diacritics:	inanim. Noun	transitive verb	inanim. noun
From inflectional endings:	nom. or acc.	3rd sing. present	nom. or acc.
English translation:	Glass	scratches	table.

Expository description: If in a Russian sentence the "current" item
is an inanimate noun having an ambiguous nom. or acc. inflectional
ending and the "immediately subsequent to current" item is a transi-
tive verb and the item immediately following this verb is an inani-
mate noun having ambiguous nom. or acc. inflectional ending, then
the "current" item is nominative (i.e., the subject) and the item
following the "immediately subsequent to current" item is accusative
(i.e., the object).

Description by symbolic logic:
Let the symbol: "aa" mean "the item is a noun"
"ab" mean "the item is a pronoun"
"ac" mean "the item is animate"
"ad" mean "the item is nominative"
"ae" mean "the item is accusative"
"af" mean "the item is a verb"

Let the symbol: "ag" mean "the item is a transitive"

"I" signify the ith item in the sentence (where "i" can have the values 1, 2, ..., n; where "n" is the number of items in the sentence.)

"." mean "and"

"v" mean "or"

"⊃" mean "if .... then"

"—" mean "not"

Then the symbolic logic statement is:

$$[aa \cdot ac \cdot (ad \ v \ ae)]_{i, (i+2)} \cdot (af \cdot ag)_{i+1} \supset ad_i \cdot ae_{i+2}$$

The flow-chart which follows is intended to show in a very general and very simplified form some of the proposals of contributors to MT and how such proposals might fit into an overall plan of MT.

The outputs of the Key-Punch (at the top of the chart) are labeled in parentheses with subscripts "H". Information so labeled is in a coded form corresponding to a pattern of punched holes. As the information emerges from the subsequent programs it is in a binary coded form.

Two types of glossary entries have been proposed.

Non-split entry:
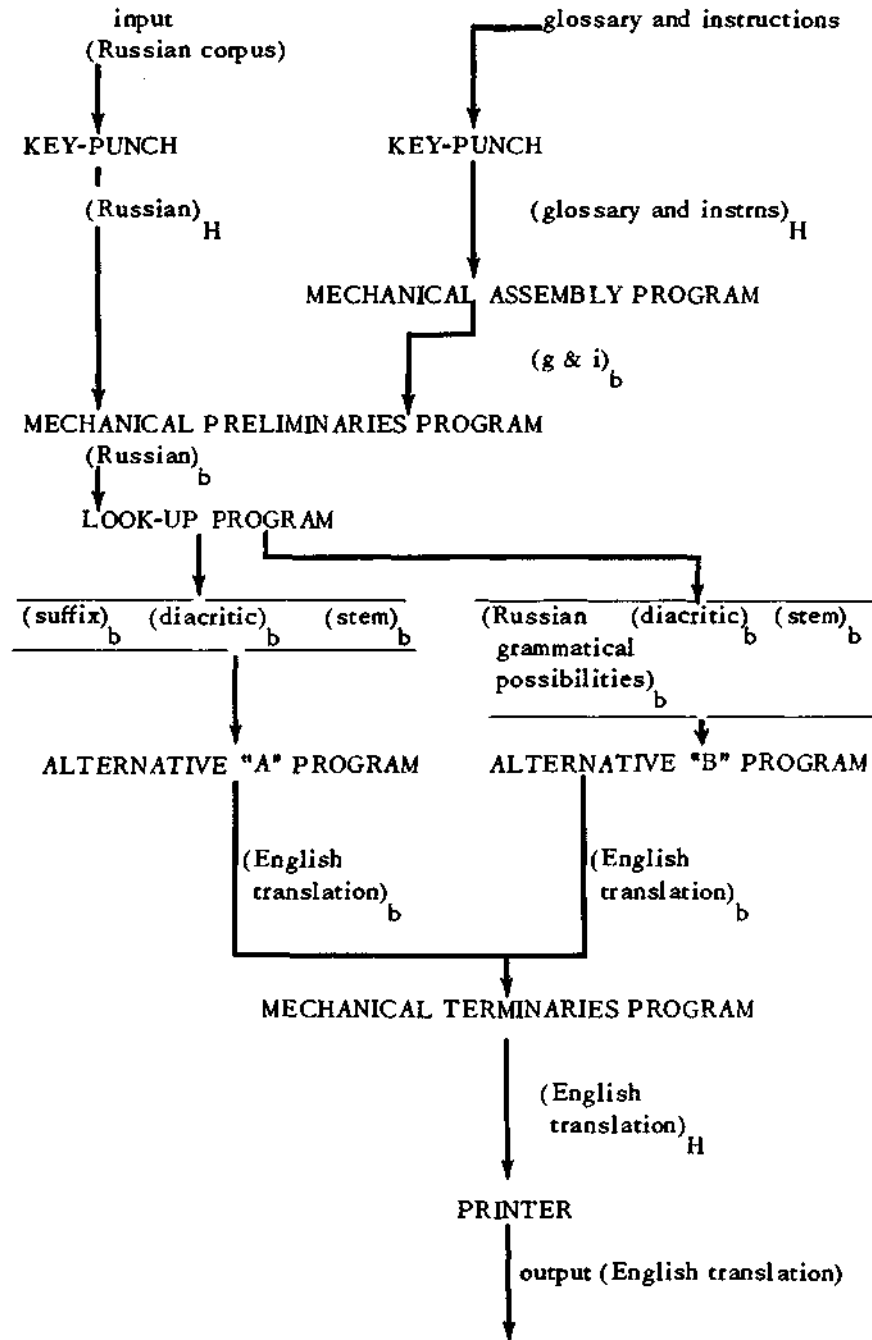Each paradigm form with its handling cue (diacritic) and with its English equivalent would constitute one entry,

Split entry:
The Russian "stem" with its handling cue (diacritic) and with its English equivalent would constitute one entry.

The same outputs of a "Look-Up" program are possible whether the glossary contains only split entries or only non-split entries or assorted split and non-split entries.

input
(Russian corpus)

glossary and instructions

KEY-PUNCH

KEY-PUNCH

(Russian)$_H$

(glossary and instrns)$_H$

MECHANICAL ASSEMBLY PROGRAM

(g & i)$_b$

MECHANICAL PRELIMINARIES PROGRAM

(Russian)$_b$

LOOK-UP PROGRAM

(suffix)$_b$ (diacritic)$_b$ (stem)$_b$

(Russian grammatical possibilities)$_b$ (diacritic)$_b$ (stem)$_b$

ALTERNATIVE "A" PROGRAM

ALTERNATIVE "B" PROGRAM

(English translation)$_b$

(English translation)$_b$

MECHANICAL TERMINARIES PROGRAM

(English translation)$_H$

PRINTER

output (English translation)

-165-

More details of Alternative "A" program:

First step:
Test for Russian words with suffix "-a"; rearrange and/or resolve lexical ambiguities and idioms and translate.

Subsequent steps:
Test for Russian words with the other suffixes and proceed as above.

More details of Alternative "B" program:

"Idiom" recognizer

↓

Russian Grammatical Ambiguity Resolver

↓

Identification of Russian Structure Type
(Russian Syntax Analysis)

↓

Rearrangement for English Structure
(English Syntax Synthesis)

↓

Lexical Ambiguity Resolver

↓

English Grammatical Identifier

↓

Look-Up
(includes English morphological synthesis)