[*International Conference on Machine Translation of Languages and Applied Language Analysis*, National Physical Laboratory, Teddington, UK, 5-8 September 1961]

A PRELIMINARY APPROACH TO JAPANESE-ENGLISH AUTOMATIC TRANSLATION *

by

SUSUMU KUNO

(Harvard Computation Laboratory)

1. FOUR STAGES OF AUTOMATIC TRANSLATION

THE proposed procedure for automatic translation of a Japanese linear text into English can be divided into four stages: 1) automatic input editing, 2) automatic segmentation with morphological analysis, 3) syntactical analysis, and 4) transformation with output editing, including semantic transfer.

2. FORMS OF INPUT TEXTS

It is apparent that input texts which are in accord with a commonly accepted writing system are better in the sense that they need less pre-editing before they are fed into a machine. Because the standard vernacular writing system of Japanese makes no use of spaces between words and because ***kanas*** (syllable Japanese characters) and ***kanjis*** (ideographic Chinese characters) are used instead of roman letters, it is necessary to devise a method of automatically cutting into its components the unsegmented sentence, written in kanas and kanjis,

In the standard writing system, 71 kanas and 1850 kanjis are used. A kana is a syllabic grapheme, which broadly speaking, corresponds to a combination of a consonant and a vowel in speech. A kanji, on the other hand, is an ideogram with one or more pronunciations attached to it. The pronunciation of kanjis, however, is usually governed by their combination with other kanjis, or with declensional kana endings. Graphemes used to represent so-called grammatical forms are in most cases kanas, but every kanji can be replaced by one or more kanas which represent its pronunciation, so that one utterance can be represented by a variety of sentences ranging from those in which kanjis are used wherever possible to those in which no kanjis are used, including a number of intermediate possibilities. In order to prevent the size of the automatic dictionary from being too much enlarged by storing two or more representations for each morpheme (when both kana and kanji representations are allowable), it is necessary to regulate the form of input kana and kanji texts. In the proposed system of automatic translation, two possibilities are considered: (a) kana texts in which no kanjis are used; and (b) kana-kanji texts in which kanjis are used wherever possible according to the official directives about the use of kanas and kanjis.[1]

---

*This study has been supported in part by the National Science Foundation.

The manual process of reducing Japanese texts to one or another of the input forms (a) and (b) will be called "pre-editing", and will be distinguished from automatic input editing, which is the first stage of the procedure for automatic translation, in which kanas and kanjis are transformed to tokens which are accepted by a computer and are convenient for subsequent automatic segmentation.

3. DICTIONARY AND AUXILIARY ITEMS

The automatic dictionary is expected to consist of dictionary items arranged in alphabetical order with various grammatical codes and English correspondents. ***Dictionary items*** are units of Japanese established for the purpose of automatic segmentation, roughly corresponding to what might be termed lexical as opposed to grammatical forms. A file of ***auxiliary items*** is to be stored apart from the dictionary file, consisting of units of Japanese corresponding roughly to so-called grammatical forms. These are to be arranged in small groups according to ***distribution types,*** also for the purpose of automatic segmentation.

All dictionary and auxiliary items are expected to be coded according to ***distribution types, function types*** and ***transformation types.*** Distribution types are categories of items established on the basis of their combination with contiguous items. Function types are categories of words established on the basis of potential roles they may fulfil in sentences; i.e., on the basis of their prediction and fulfillment of syntactical relationships. Units of syntactical analysis are called ***words.*** They do not necessarily correspond to items, which are essentially units of automatic segmentation. A word consists of one or more items, and the function type of a word is a product of the function type codes of its component items. For instance, if the dictionary item "hanas" ("to speak"), having a function type code for ***verb,*** is combined with the auxiliary item "u" (verbal final-attributive suffix), which has a function type code for ***final,*** the function type of the resulting word "hanasu" will be ***final verb FT.*** Although the distribution type of an item seems to be rather closely connected with its function type code, it is necessary to distinguish between the two. In order to avoid confusion in terminology, names of distribution types and function types will always end with ***DT*** (distribution type) and ***FT*** (function type) respectively; e.g., ***substantive DT,*** and ***substantive FT.*** ***"DT substantives"*** and ***"FT substantives"*** refer to members of the ***substantive DT*** and ***substantive FT*** categories. Transformation types are categories of words established on the basis of their roles in the structural transfer between two languages, pertaining mainly to word order, omission of words in the source language, and insertion of new words in the target language.

4. AUTOMATIC INPUT EDITING

In the first stage of the automatic language translation process, each kana in an input text will be transformed into two tokens for two roman letters so as to preserve a one-to-one correspondence between kanas and their correspondent roman letters. In a kana-kanji input text,

however, each kanji will be transformed into an irreducible unit token. For Instance, three kanas (shown in entry 1, *Table 1)* in a kana text will be replaced by tokens for "hanasu", which has six characters: "h", "a", "n", "a", "s", and "u". On the other hand, a kanji and a kana (entry 2, *Table l)* in a kana-kanji text will be replaced by tokens for "(hanasu", which has three characters: "(hana)", "s" and "u". The replacement of each kana by two reduced tokens in a kana and kana-kanji text is due to the assumption that the Japanese inflectional system is better analyzed on the level of roman letters than on the level of irreducible kanas, with fewer varieties of suffixes and fewer rules of permissible combinations with canonical stems, and with fewer possibilities of homographic verbal stems. Replacement of each kanji by one irreducible token, on the other hand, is due to the expectation that in the prospective analysis no kanji will contain more than one "morpheme".

The above mentioned transformation may be done automatically by means of a kana typewriter or a kana-kanji typewriter equipped with magnetic tape or other memory devise for internal conversion to the desired representation. Input provisions will vary according to the type of computer used.

TABLE 1

Kana and Kanji Reference

Entry	Kanas and Kanjis
1	ハナス
2	話ス

5. AUTOMATIC SEGMENTATION

The second stage of the process pertains to the automatic segmentation of a continuous run of tokens for representations of kanjis and kanas. The proposed method of automatic segmentation is based on the prospect that in our analysis auxiliary items will be shorter in length and fewer in number than dictionary items, and that no problem will be caused by assuming that every "phrase" in a sentence begins with a dictionary item whose average length is greater than that of auxiliary ones, or by Including "prefixes" in the category of dictionary items, as they are very scarce in Japanese.

A method is proposed providing for a "find the longest matching dictionary item" (subsequently referred to as "find the longest") operation combined with the testing of immediately following sequences of tokens against predicted auxiliary items (referred to as "predictive testing"). The distribution type of the longest matching item is examined, and then a string of tokens immediately following a "matched segment" (i.e., a segment corresponding to a dictionary item found by the previous operation) is tested to determine whether it is initiated by any of the auxiliary items which are predicted on the basis of that distribution type.

Suppose the distribution type of the longest matching item found at the beginning of a text is *substantive DT* which is assumed, as a simplified model, to allow nothing to follow except one or more *DT particles.* This item is first considered to be *relevant* on the basis that every dictionary item can be combined with a preceding space, actual or hypothetical. The next step is to go to a subroutine in which each *DT particle* predicted to succeed this distribution type is tested against a string of tokens immediately following the matched segment. If a segment or segments matched by one or more *DT particles* are found, they are separated, and it is assumed that each matching item, with the exception of the last *DT particle* found, is *locally valid* on the basis that it is followed by an item whose combination with it is permissible in the language. Then comes a new "find the longest" operation, and the longest matching item, if found, is tested against the item last separated to determine whether or not the combination of the two is permissible. If it is permissible, the newly found item is said to be relevant, and the preceding item is said to be locally valid; if it is not, the newly found longest matching item is first considered to be *irrelevant* and the second longest matching item is then sought.

If, on the contrary, no segment matched by any of the *DT particles i*s found following the first *DT substantive,* the dictionary item is tested as to whether it can be followed directly by another dictionary item with no intervening auxiliary item(s). If the answer is yes, a new "find the longest" operation is performed upon the remainder of the sentence, and the matching item found is used as a key for determining the local validity of the preceding item. If the answer is no, the first item *(DT substantive)* is considered to be *invalid,,* and the second longest matching item is sought.

When the whole sentence has been cut into segments successfully matched by locally valid items, it is assumed that these items are *wholly valid* on the basis of the proposed program of automatic segmentation. Both structurally and semantically valid Items are said to be *correct.*

One or more matched segments in a sentence beginning with a dictionary item and ending with an auxiliary item (if any) immediately before the next dictionary item will be said to form a *joint,* whose *nucleus* is a segment matched by a dictionary item, and whose *subsidiary* is a segment matched by an auxiliary item. Joints are classified according to the distribution type to which their nuclei belong. The longest possible combination of a nucleus and subsidiaries in each type of joint is called a *maximum joint.* Rules of the combination and ordering of auxiliary items as subsidiaries in maximum joints are studied in detail. Auxiliary items are classified according to their relative ordering in a joint; e.g., as to which elements must precede, or are prohibited from following others, etc.

5.1 Inclusion Marks

These are various ways of programming the procedure to be utilized in automatic segmentation. The technical problem will not be discussed in detail, but a brief outline of a proposed method follows.

When a machine with a large addressable memory is available, it is comparatively easy to incorporate the "find the longest" instruction in a table look-up process. There may be cases in which the longest matching

item found is not correct, because during the operation the proper item has been erroneously associated with a string of letters immediately following (which may or may not be meaningful). For example, in automatically segmenting "hanayaki", "hanaya" ("flower shop") will be selected as the longest matching item for an automatic dictionary which has entries such as those shown in *Table 2.* But this segmentation is wrong because in the above text, a cut should most probably be made between "hana" ("flower") and "ya" *(particle DT* "and") and "ki" ("tree"), on the assumption that "hanaya" will never be followed by any item beginning with ("ki". Likewise, "hanakago" as one unit means "flower basket", but it may also be "hana ka go", "hana" meaning "flower" and "ka" being a *DT particle* meaning "or". The final segment, "go" may either mean "five' or the game of "go", or may constitute a nonsensical sequence of letters detached from the beginning of an item with a form such as "goma" ("sesame"), "gomi" ("dust"), or "gobo'u" ("burdock").

To prevent an erroneous segmentation of this kind, what may be called an "inclusion mark" may be prepared for every dictionary and auxiliary item. This mark consists of a single digit indicating where an alternative cut may be made (counting backwards in the sequence of letters), and showing the dictionary location of the shorter item thence produced (see Table *2).* A cut is made only when both of the resultant two segments are legitimate. "Hanawa", for instance, has an inclusion mark "0" on the assumption that "hana" will never be followed by any item beginning with "wa".

TABLE 2
Illustration of Inclusion Marks

Address	Entry	Inclusion Mark		English Correspondent
al	ha	0		tooth, leaf
•				
•				
•				
b1	hana	2,	al	flower, nose
b2	hanabana	0		flowers
b3	hanabanasi	2,	b2	brilliant
b4	hanabi	0		fireworks
b5	hanakago	4	bl	flower basket
•				
•				
•				
cl	hanas	1	bl	to talk
c2	Hanasi	1	cl	story
c3	hanataba	0		bouquet
c4	hanawa	0		wreath
c5	hanaya	2,	b1	flower shop
c6	hanayaka	2,	c5	gay, splendid
c7	hanayome	0		bride

5.2 Program of Automatic Segmentation

Figure 1 contains a flow chart for finding the longest matching item in the dictionary and for predictive testing of auxiliary items, with previously made segmentation corrected on the basis of Inclusion marks. The program is divided into three blocks. Block A is for normal repetition of cycles for "finding the longest" and succeeding "predictive testing"; block B is for rejecting the longest matching item and taking a shorter one as indicated by the inclusion mark; block C is for correcting retrospectively segmentation previously made. An explanation of the notations used in the flowchart is given below.

a. Steps in the operation are numbered A1, A2,...; B1, B2,...; etc., the initial roman letters indicating block numbers.

b. "combination allowed?" (A2, B3, C6) means "Is the combination of a newly found item with a previously found item allowed?" If the answer is yes, the newly found item is considered to be relevant, and the preceding item to be locally valid. If the answer is no, the newly found item is considered to be irrelevant. Items found through "predictive testing" (A4) do not undergo this operation since it is clear that they are relevant.

c. Punctuation marks and spaces are assumed to be included in auxiliary items.

d. Segments in the text which are matched by dictionary and auxiliary items found by the "find the longest" (A1 and B2) and "predictive testing" (A4) operations respectively are to be separated and stored in temporary storage in the order of their text occurrence.

e. "Form a joint by itself?" (B1) means "Can the item form a joint by itself, without being followed by any auxiliary item?"

f. "Find the longest immediately following "(B8) means "Find the longest matching item at the head of the remaining text."

g. "Take shorter item" (B6, B8, C5) means "Bring into register, and place in temporary storage the shorter item as indicated by the inclusion mark of the item found through operation of steps B2, A1, and C1 respectively, and modify the remaining text".

H. "Take previously separated segment"(C1) means "Bring into register a foregoing previously separated segment, taking it out of temporary storage and at the same time returning it to the remaining text."

1. If "end of sentence" (A5) is reached, all the previously found items are considered to be wholly valid.

J. The missing word routine has not yet been studied.

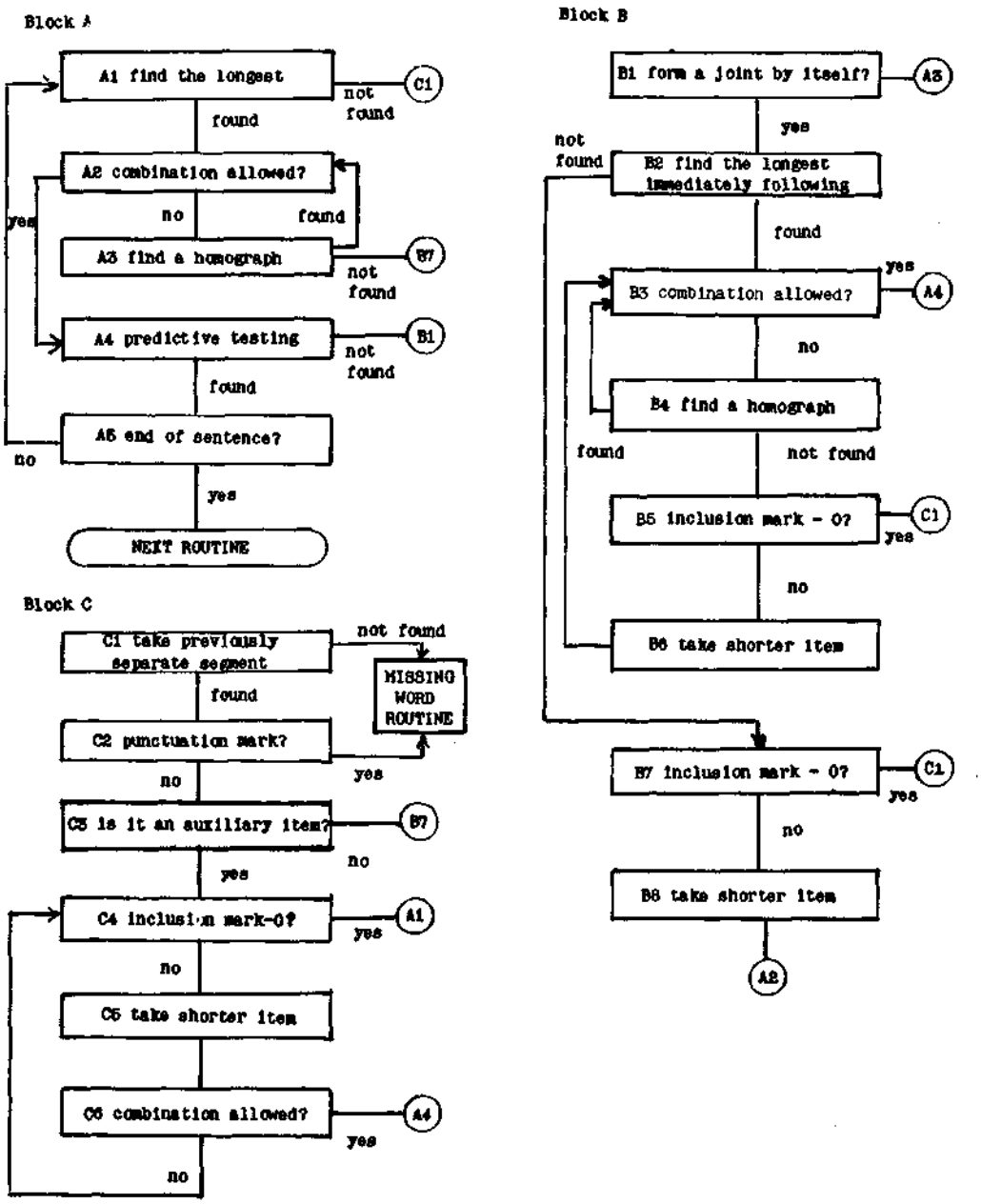

Fig. 1 Flowchart of Automatic Segmentation

Figure 2 shows the process of automatic segmentation of *"sorehahana-yaka'u'e kiyani'aru".* The "step numbers" in the first column correspond to the step numbers in the flowchart of *figure 1.* Matched segments and tested inclusion marks are shown in the fourth column under the heading "register". Matched segments are separated and stored in "temporary storage" (fifth column). If nothing is stored in the "register", "temporary storage", or "remaining text", a "Δ" is used.

Correct segmentation of the input sentence will yield "sore" *(substantive DT* "it")/ "ha" (thematic *particle DT)/* "hanaya" *(substantive DT* "flower shop")/ "ka" *(particle DT* "or")/ "'u'ekiya" *(substantive DT* "gardener's")/ "ni" *(particle DT* "at")/ "'ar" *(consonantal verbal stem DT* "to be")/ "u" *(consonantal verbal suffix DT)/* "." *(period DT).* Difficulty may be expected because it is probable that stored in the dictionary are both an item longer than "hanaya", that is "hanayaka" *(adjectoverb DT* "gay, splendid"), having the inclusion mark "2" and the address of "hanaya" (see *Table 2),* and an item longer than "'ar", that is "'aru" *(joint former DT* "a certain, unnamed") that belongs to the distribution type which can form a joint by itself, but which never stands before a period.

In the process of automatic segmentation of this sentence, "hanayaka" Is considered to be relevant because its combination with the preceding item "ha" is permissible. It is then found that "hanayaka" is not followed by any predicted auxiliary items, that it cannot form a joint by itself, and that it has no homographs. Following the "inclusion mark - 0?" and "take shorter item" operations, the shorter item "hanaya" is chosen, and the automatic segmentation continues correctly.

The input sentence in *figure 3* is *"hanayahaga'utukusi'i".* Correct segmentation will be "hana" *(substantive DT* "nose, flower")/ "ya" *(particle DT* "and")/ "ha" *(substantive DT* "tooth, leaf")/ "ga" *(subjective particle DT)/* "'utukusi " *(Adjectival stem DT* "beautiful")/ "'*i" (adjectival suffix DT,* present final-attributive)/ "." *(period DT).* Difficulty may be caused if an item is stored in the dictionary which is longer than "hana"; that is "hanaya" (see T*able 2),* and if the local validity of "hanaya" is assumed by a following "ha" (thematic *particle DT),* and it may also be caused if the next "find the longest" operation yields "ga" *(substantive DT* "month"), on the basis of which the preceding "ha" is assumed to be locally valid. It is at this point of segmentation, when "hanaya ha ga" is reached, that it comes to a dead end, which makes possible correction of the previous made segmentation. "Ga" *(substantive DT),* however, is not followed by any predicted auxiliary items, and cannot form a joint by itself, Since "ga" is not found to be locally valid, the routine for correcting the previously made segmentation "hanaya ha ga" is initiated.

In each example, the entire sentence has been cut into wholly valid segments, which match exactly correct segments of the sentence.

step number	operation	fulfilled condition	register	temporary storage	remaining text
A1	find the longest	FOUND	sore	sore	hahanayaka'u'ekiyani'aru.
A2	combination allowed?	YES		〃	〃
A4	predictive testing	FOUND	ha	sore ha	hanayaka'u'ekiyani'aru.
A5	end of sentence?	NO		〃	
					〃
A1	find the longest	FOUND	hanayaka	sore ha hanayaka	'u'ekiyani'aru.
A2	combination allowed?	YES		〃	〃
A4	predictive testing	NOT FOUND	Δ	〃	〃
B1	form a joint by itself	NO		〃	〃
A3	find a homograph	NOT FOUND	Δ	〃	〃
B7	inclusion mark-0?	NO	2	〃	〃
B8	Take shorter item		hanaya	sore ha hanaya	ka'u'ekiyani'aru.
A2	combination allowed	YES		〃	〃
A4	predictive testing	FOUND	ka	sore ha hanaya ka	'u'ekiyani'aru.
A5	end of sentence?	NO		〃	〃
A1	find the longest	FOUND	'u'ekiya	sore ha hanaya ka 'u'ekiya	ni'aru.
A2	oombination allowed?	YES		〃	〃
A4	predictive testing	FOUND	ni	sore ha hanaya ka 'u'ekiya ni	'aru.
A5	end of sentence?	NO		〃	〃
A1	find the longest	FOUND	'aru	sore ha hanaya ka 'u'ekiya ni 'aru	
A2	combination allowed?	YES		〃	〃
A4	predictive testing	NOT FOUND	Δ	〃	〃
B1	form a joint by itself?	YES		〃	〃
B2	find the longest, immediately following	NOT FOUND	Δ	〃	〃
B7	inclusion mark-0?	NO	1	〃	〃
B8	Take shorter item		'ar	sore ha hanaya ka 'u'ekiya ni 'ar	u.
A2	combination allowed?	YES		〃	〃
A4	predictive testing	FOUND	u	sore ha hanaya ka 'u'ekiya ni 'ar u	.
		FOUND	.	sore ha hanaya ka 'u'ekiya ni 'ar u .	Δ
A5	end of sentence?	YES			
	↓ NEXT ROUTINE				

Fig. 2 Example 1 (Automatic Segmentation)

step number	operation	fulfilled condition	register	temporary stage	remaining text
A1	find the longest	FOUND	hanaya	hanaya	haga'utukusi'i.
A2	combination allowed?	YES		"	"
A4	predictive testing	FOUND	ha	hanaya ha	ga'utukusi'i.
A5	end of sentence?	NO		"	"
A1	find the longest	FOUND	ga	hanaya ha ga	'utukusi'i.
A2	combination allowed?	YES		"	"
A4	predictive testing	NOT FOUND	Δ	"	"
B1	form a joint by itself?	NO		"	"
A3	find a homograph	NOT FOUND	Δ	"	"
B7	inclusion mark-0?	YES	0	"	"
C1	take previously sep. seg.	FOUND	ga	hanaya ha	ga'utukusi'i.
C2	punctuation mark?	NO		"	"
C3	is it an auxiliary item?	NO		"	"
B7	inclusion mark-0?	YES	0	"	"
C1	take previously sep. seg.	FOUND	ha	hanaya	haga'utukusi'i.
C2	punctuation mark?	NO		"	"
C3	is it an auxiliary item	YES		"	"
C4	inclusion mark-0?	YES	0	"	"
A1	find the longest	FOUND	ha	hanaya ha	ga'utukusi'i.
A2	combination allowed?	NO		"	"
A3	find a homograph	NOT FOUND	Δ	"	"
B7	inclusion mark-0?	YES	0	"	"
C1	take previously sep. seg.	FOUND	ha	hanaya	haga'utukusi'i.
C2	punctuation mark?	NO		"	"
C3	is it an auxiliary item?	NO		"	"
B7	inclusion mark-0?	YES	0	"	"
C1	take previously sep. seg.	FOUND	hanaya	Δ	hanayahaga'utukusi'i.
C2	punctuation mark?	NO		"	"
C3	is it an auxiliary item?	NO		"	"
B7	inclusion mark-0?	NO	2	"	"
B8	take shorter item		hana	hana	yahaga'utukusi'i.
A2	combination allowed?	YES		"	"
A4	predictive testing	FOUND	ya	hana ya	haga'utukusi'i.
A5	end of sentence?	NO		"	"
A1	find the longest	FOUND	ha	hana ya ha	ga'utukusi'i.
A2	combination allowed?	YES		"	"
A4	predictive testing	FOUND	ga	hana ya ha ga	'utukusi'i.
A5	end of sentence?	NO		"	"
A1	find the longest	FOUND	'utukusi	hana ya ha ga 'utukusi	'i.
A2	combination allowed?	YES		"	"
A4	predictive testing	FOUND	'i	hana ya ha ga 'utukusi 'i	.
A5	end of sentence?	FOUND YES	.	hana ya ha ga 'utukusi 'i .	Δ
	NEXT ROUTINE				

Fig. 3 Example 2 (Automatic Segmentation)

There would be less difficulty in automatic segmentation if the above examples were kana-kanji texts in which tokens for the two kanjis and the kana, "(hana) (ya)ka" ("flower shop" and "or"), (see entry 1, *Table 3),* were different from those for the four kanas (entry 2), "hanay-ka" ("gay, splendid"); tokens for the kanji and kana (entry 3), "('a)ru" ("am, is, are"), from those for the two kanas of entry 4, "'aru" ("a certain, unnamed"); and tokens for the kanji and kana of entry 5, "(hana)ya" ("flower" and "and"); from those for the two kanjis, "(hana) (ya)" ("flower shop"), (entry 6). We no longer have an option for arriving at an item longer than the correct one. Generally speaking, automatic segmentation of kana-kanji texts seems to be far easier than that of kana texts.

TABLE 3
Kana and Kanji Reference

Entry	Kanas and Kanjis
1	花屋カ
2	ハナヤカ
3	有ル
4	アル
5	花ヤ
6	花屋

6. SYNTACTICAL ANALYSIS

The method of syntactical analysis proposed is that of predictive analysis, originally conceived by Rhodes[2], adopted and developed at Harvard University for Russian by Sherry[3] in collaboration with Oettinger, for English by Bossert, Giuliano and Grant[4], with theoretical implications of the method having been investigated by Oettinger [5,6].

One peculiarity of predictive analysis as applied to Japanese is that it seems more convenient to start sentence analysis from the end of a sentence. This is based on the expectation that words having a final position in a sentence are extremely limited, being confined as a rule to the function type classes *PT final verbs, FT final adjectives* and *FT final copulas* which offer more information about the structure of a sentence than do those occurring initially. Moreover, it seems that particles which show case, prepositional or conjunctional relationships always follow words, phrases or clauses to which they are attached, and that attributive words, phrases and clauses always stand before *DT substantives* which they modify. A complete description of the function

types and essences recognized in the planned experimental system will not be given in this paper, though they are mentioned briefly in the following example for the purpose of illustrating the proposed method of syntactic analysis.

In the course of going through a sentence, predictions are constantly being generated and tested for fulfillment. As an example, let us take "nezumiganekowo(koro)sita(hanasi)ha(watakusi)wo('odoro)kaseta". ("the story that a rat killed a cat surprised me.") Let us suppose that it has been correctly segmented through stage two of the program of automatic translation and separated into component words with their function types identified.

11.	"nezumi"	*substantive FT*	("rat")
10.	"ga"	subjective *particle FT*	
9.	"neko"	*substantive FT*	("cat")
8.	"wo"	objective *particle FT*	
7.	"(koro)sita"	*final verb FT,* takes "wo" as *objective marker*	(killed")
6.	"(hanasi)"	*substantive FT,* can take a clause of apposition	("story")
5.	"ha"	thematic *particle FT*	
4.	"(watakusi)"	*substantive FT*	("I")
3.	"wo"	objective *particle FT*	
2.	" ('odoro)kaseta"	*final verb FT,* takes "wo" as *object marker*	("surprised")
1.	"."	*period FT*	

The procedure of syntactical analysis, somewhat simplified, is the following:

a. First of all, the prediction of 1, *end of sentence, i*s stored in the prediction pool.

b. The first item to be brought into the register is a period, which fulfils prediction 1, and wipes it. It will generate prediction 3, *final particle essence,* and 2, *predicate head,* the former being placed at the top of the prediction pool, the latter at the bottom. These two predictions have what is called an "exclusion wipe mark", which causes all the predictions made by the same word to be wiped If any one which has the mark has been fulfilled.

c. The second word ("('odoro)kaseta") fulfills prediction 2, wipes both predictions in the pool, and in turn predicts *5, object marker-A* (to be fulfilled by "wo") and 4, *subject marker.*

d. The third word ("wo") fulfills prediction 5, wipes it, and in turn predicts 6, *object master.* The content of the prediction pool is now
6. *object master.*
4. *subject marker.*

e. The fourth word ("(watakusi)") fulfills prediction 6, wipes it, and predicts 10, *attributive substantive essence, 9, attributive phrase*

marker, 8, attributive adjective essence, and 7, *relative predicate head. Relative predicate head* is an essence to be accepted by the so-called final-attributive forms of verbs, adjectives and copulas which modify succeeding nouns. Now the content of the pool is

10. *attributive substantive essence*
9. *attributive phrase marker*
8. *attributive adjective essence*
7. relative predicate head
4. *subject marker*

f. The fifth word ("ha") fulfills prediction 4, which has what is called an "endwipe mark" which causes all the preceding predictions to be removed. The fifth word itself generates a prediction of 11, *subject master.* The content of the prediction pool is now only

11. *subject master.*

g. The sixth word ("(hanasi)") fulfills prediction 11, and in turn predicts 15, *attributive substantive essence. 14, attributive phrase marker,* 13, *attributive adjective essence,* and 12, *relative predicate head.*

h. The seventh word ("(koro)sita") fulfills prediction 12, and predictions 15, 14, 13 and 12 will be wiped due to the exclusion wipe mark accompanying the *relative predicate head* prediction. Predictions newly made are 17, *object marker-A* (to be fulfilled by "wo"), and 16, *relative subject marker.* They do not have the exclusion wipe mark since the sixth word ("(hanasi)") can take an attributive clause of apposition, so that both predictions may be fulfilled. (If "(watakusi)" instead of "(hanasi)" occurs, for instance, the two predictions will have the exclusion wipe mark since "(watakusi)" belongs to a class of substantive function types which cannot take any attributive clause of apposition, method, reason, time or place, and the relative to be inserted must be a relative pronoun of either subject or object.) The content of the prediction pool is now

17. *object marker~A*
16. *relative subject marker .*

1. The eighth word ("wo") fulfills prediction 17, wipes it, and generates a new prediction, 18, *object master.* The content of the pool is now

18. *object master*
16. *relative subject marker.*

j. The ninth word ("neko") fulfills prediction 18, wipes it, and generates new predictions 22, *attributive substantive essence,* 21, *attributive phrase marker,* 20, *attributive adjective essence,* and 19, *relative predicate head.* Below these predictions is prediction 16, *relative subject marker.*

k. The tenth word ("ga") fulfills prediction: 16, and reaching the "endwipe mark", wipes all the preceding predictions in the pool together with it. It generates a new prediction 23, *subject master.*

1. The eleventh and final word, "nezumi", fulfills prediction 23 and concludes the syntactical analysis of the sentence.

In the process of analysis, each word in a sentence will be assigned a) an essence which has been fulfilled by it, b) a linkage number which shows by which word it has been predicted, and c) a group number which shows to which clause in the sentence it belongs (see *Table 2).*

One syntactical peculiarity of Japanese is that, unlike English, the subject of a sentence is very often omitted. Some provision must be made for insertion of subjects in English sentences when necessary. In the proposed system of predictive analysis, fulfillment of the *subject marker* and *relative subject marker* predictions is regarded as essential. If no Japanese words are found which fulfill these predictions, they are transferred together with the *subject master* essence, to a fulfilled essence pool with a mark to show they have no input counterpart, before they are wiped by an "endwipe mark" or after all words constituting a sentence have been tested.

TABLE 2

Clause Level Designation of Essences Fulfilled

Word Number	Word	Essence Fulfilled	Linkage Number	Group Number
11	nezumi	subject master	10	2
10	ga	relative subject marker	7	2
9	neko	object master	8	2
8	wo	object marker	7	2
7	(koro)sita	relative predicate head	6	2
6	(hanasi)	subject master	6	1
5	ha	subject marker	2	1
4	(watakusi)	object master	3	1
3	wo	object marker	2	1
2	('odoro)kaseta	predicate head	1	1
1	.	end of sentence	INIT	0

Objects of verbs are often omitted in Japanese, but it is not necessary to supply them since objects of verbs are frequently omitted in English.

7. TRANSFORMATION WITH OUTPUT EDITING

Stage four of the program of automatic translation deals with the synthesis of the target language, in which word-order transformation is a serious problem. In brief, words which have the same group number are gathered together and within each group, transformation of word order is performed.

Subject marker, object marker, and *relative subject marker* are omitted. *Subject master* or *relative subject master* comes first within each group, followed by *predicate head* or *relative predicate head,* and then by *object master.* For the *subject master,* which has no Japanese counterpart, an imaginary substantive "X" is introduced, which has English correspondent "X".

Groups 1 and 2 of the above example will be arranged as follows:

Group 1

6.	(hanasi)	*subject master*
2.	('odoro)kaseta	*predicate head*
4.	(watakusi)	*object master*

Group 2

11.	nezumi	*relative subject master*
7.	(koro)sita	*relative predicate head*
9.	neko	*relative object master*

Group 2 will be inserted immediately following "(hanasi)" (Group 1), with a conjunction of apposition "that". The inflected English correspondents will be approximately as follows:

6	story
-	that
11	rat
7	killed
9	cat
2	surprised
4	me
1	.

8. CONCLUSION

The results of preliminary manual testing of automatic segmentation on the basis of a "find the longest matching dictionary item" operation followed by "predictive testing" has given reason to believe that this program will provide a practical basis for the analysis of running kana-kanji text. Thirty-nine distribution types for Japanese have thus far been recognized, but no exhaustive classification of dictionary and auxiliary items into these types has been attempted. In particular need of further study are the problems of homographs and missing words.

Function types and essences are now under study, and experimental sentence analysis has indicated that predictive analysis should provide

an effective method of obtaining the more probable analysis for a given input sentence on a single right-to-left pass. The right-to-left pass proposed, entailing as it does analysis proceeding in a direction converse to transcription, raises an important question about the syntactical nature of Japanese, and about Miller's and Yngve's hypotheses on the mechanism of temporary memory in humans [7,8]. This is a problem worthy of serious consideration.

Since the system proposed in this paper has neither been developed in complete detail nor been tested on a machine, it will be subject to various improvements as the system is further refined.

REFERENCES

1. Toyo-Kanjis, "Chinese characters for dally use in Japanese, promulgated by Japanese Government in 1946 on the basis of the decision and report of Japanese Language Council; "Table of Pronunciations of Toyo-Kanjis," Japanese Government, 1948".

2. RHODES, I., "A New Approach to the Mathematical Translation of Russian", National Bureau of Standards, Washington, D.C., unpublished report, 1959

3. SHERRY, M.E., "Syntactic Analysis in Automatic Translation", Doctoral Thesis, Harvard University, 1960.

4. BOSSERT, W., GIULIANO, V. and GRANT, S. "Automatic Syntactic Analysis of English," *Mathematical Linguistics and Automatic Translation,* Report No. NSF-4, Harvard Computation Laboratory, Section VII, 1960.

5. OETTINGER, A.G., "Current Research on Automatic Translation at Harvard University", National Symposium on Machine Translation, Los Angeles 1960, (to appear in ***Proceedings*** of the Symposium, Prentice-Hall, Englewood-Cliffs, New Jersey).

6. OETTINGER, A.G., "Automatic Syntactic Analysis and the Pushdown Store", Symposium on the Structure of Language and its Mathematical Aspects, 567th Meeting of the American Mathematical Society, New York 1960, (to appear in ***Proceedings*** of the Symposium, American Mathematical Society, Providence, Rhode Island).

7. MILLER, G.A., "Human Memory and the Storage of Information", I.R.E. *Transactions on Information Theory,* 1956, **IT-5**, pp.129-137.

8. YNGVE, V.H., "A Model and an Hypothesis for Language Structure", *Proc. Amer. Phil. Soc.,* 1960, **104,** No. 5 pp. 444-466.