A Principle-based Korean/Japanese Machine Translation System: NARA

Hee Sung Chung
Electronics and Telecommunications Research Institute
P.O.Box 8, Dae-Dog Danji, Chung-Nam, Republic of KOREA

ABSTRACT

This paper presents methodological and theoretical principles for constructing a machine translation system between Korean and Japanese. We focus our discussion on the real time computing problem of the machine translation system. This problem is characterized in the time and space complexity during the machine translation. The NARA system has the real time computing algorithm which is based on a mathematical model integrating the linguistic competence and the linguistic performance of both languages, with consequence that the system NARA has also the functional characteristic: the two-way translation mechanism.

1. Introduction

We are developing a two-way (bidirectional) simultaneous Korean/Japanese machine translation: NARA[7]. The NARA system is designed by a specific computing model, which is a mathematical model based on the methodological and theoretical principles involving the formalization of the two-way simultaneity. The most significant characteristics of using the formal description for the NARA system is that the descriptive contents of representative algorithm do not depend upon the conventional approaches to machine translation. They have only methodological and theoretical arguments such as transfer method and pivot method, and adopt linguistic theory for language model with an ad hoc manner.

In other words, current approaches to machine translation are usually focusing on the engineering feasibility; therefore they explain only what kinds of data structure of the language they employ, how they are analyzed and how they are translated. They do not give the details of their capability and limitation for the methodology of the machine translation system.

The aim to develop the machine translation system is to translate an enormous amount of information written in a foreign language. In this purpose, it is said that the translation is changing from an art to a technique. If we give a clear-cut answer to the argument that the translation is the technique, it is natural that the translation merely means the physical transference of the contents of the language. To realize the idea for mechanizing the translation process, we need to formalize the translation mechanism. In this paper, we propose a methodology needed for the improving the quality and quantity of a machine translation system.

This work is partially supported by the Korea Telecommunication Authority (KTA) under Grant 8IBT010.

2. The general principles of the NARA system

In the NARA system we take a methodological principle into consideration on the general and specific aspects for the machine translation system. The former is the hypothesis for constructing the machine translation system. The latter is the computational model that applies the general principles to the NARA system. The computational model for the NARA system is constructed by the principles of computing theory that produce a vital link between what and how. *What* means the linguistic knowledge for constructing machine translation system and *How* means the procedure that maps the collection of inputs to the desired outputs.

Our approach is intuitively motivated by Chomsky's hypothesis [4]: homogeneous communication by the same linguistic performance is possible among those who have the same linguistic competence. The *linguistic performance* means the *real time* processing of the language, and the *linguistic competence* means the knowledge of a language. The performance theory can not be developed without the competence theory.

This hypothesis suggests that mutual communication is possible among different human language systems. Thus we may represent the above concept as follows:

the description of mutual communication environment = the description of linguistic competence + the description of linguistic performance.

And we may analogously represent the concept of two way simultaneous translation as follows:

the description of two way simultaneous translation = the description of knowledge of both languages + the description of performance knowledge of the both languages.

This schema can be expanded a step further to be:

a two way simultaneous translation algorithm = the model of the corresponding data structure of the source language and the target language + the model of real time processing.

A key point of contact between the theory of grammar and the translation control is the natural link between the theory of knowledge representation and the theory of knowledge processing for the machine translation system.

We define the knowledge representation and the knowledge processing for machine translation system as a competence model and as a performance model, respectively. The competence model consists of the various kind of linguistic knowledge: morphology, syntax and semantics for the NARA system, and the performance model consists of several subareas. The first is concerned with which knowledge representations are constructed during simultaneous translation; the second is concerned with how the representations are utilized during translation; the third is concerned with the measure of computational complexity during translation. We presume that these three components constitute a complete computational model for the machine translation mechanism: a knowledge representation, an algorithm and a complexity.

We summarize the following items as the subjects of the general principle for our computational approach of the NARA system.

(1) The theory of common grammar:

we are requiring a common grammar to be suitable for the description of both languages. The common grammar is similar to the significance such that modern linguistic theory interprets the theory of universal grammar (UG) as part of a theory of language acquisition[3]. whereby we adopted a unification-based grammar formalism: K-J(J-K) grammar as a common grammar based on the correspondence existing between both languages.

(2) The notion of direct realization of translation:

as mentioned above, in order to guarantee two-way simultaneous translation, we identify the rules of the grammar with the manipulative units of translation in a one-to-one fashion. This notion is based on the grammatical covering, type transparency, grammar modification and invariants of formal language [10].

(3) The notion of complexity measure:

the complexity of the algorithm, which is the direct association between the cost time and the sequential operation during translation, should be measured.

(4) The notion of translation results:

we compare our translation results to Thorn's hypothesis: a principle of isomorphism [12].

3. The specific principles of the NARA system

The NARA system adopts several specific and theoretical principles and they are described in the following.

(1) Equivalence of grammar:

only if two grammars generate the same sets of surface sentences, they are weakly equivalent. In addition, if the two grammars generate the same language by means of the same two structure (here, by a one-to-one correspondence of rule steps), the two grammars are strongly equivalent [10]. Paraphrasing it, grammar G_1 and G_2 are weakly equivalent if the string language generated by G_1, $L(G_1)$, is identical to that of G_2, $L(G_2)$. If G_1 and G_2 are strongly equivalent, G_1 and G_2 can assign the same structural description for each word in $L(G_1)$ and $L(G_2)$. We apply this notion to the correspondence between Korean and Japanese.

(2) Grammar covering and grammar modification:

intuitively, a grammar is said to *cover* another if the first grammar can be used to easily recover all the parse structure that the second grammar assigns to an input sentence. In other words, grammar covering means that the first grammar can be used instead of the second grammar to parse a sentence of the language generated by the second grammar. This grammatical covering relation is easy to understand from the mere fact that we use the first language to study the second language. More importantly, one of the two grammars can serve as the true competence grammar for a language because it generates the proper structural description. The reason for using this principle is that the covering grammar may be more suitable for the efficiency of the processing in terms of time and space, and if a grammar covers another, the semantic rule for translation between both languages can be used to pair exactly the same input string and its meaning.

(3) Type transparency:

in our view, the type transparency is the relationship between a covering grammar and the operation units of translation. From the usual linguistic claim that a more compact grammar is more easily processed, we impose the condition that the logical organization of rules and the structure incorporated in a grammar may be mirrored exactly in the mechanism of translation.

According to our theoretical and specific principles, we can represent the structural description of translation processing, and then apply a simple mapping to the translation mechanism. This mapping is from a parse tree to a parse tree.

4. The competence model of the NARA system

In this section, we focus our attention on the concrete language knowledges such that what kinds of linguistic description are used. In order to investigate the correspondence between both languages, we partition a grammar into components: segmented word, word order, morphology, syntax and semantics. The hierarchical separation of a grammar constitutes an important step in the modularization of a translation subsystem.

4.1. Morphology

The study of the structure of words, occupies an important place within the competence model, sandwiched as it is between phonology and syntax. Morphemes may also be partitioned into lexical and grammatical classes. In both languages, lexical morphemes are generally free, while many of the grammatical morphemes are bound.

In a given Korean-Japanese/Japanese-Korean dictionary, let D_k be the set of morphological words of Korean and D_j be the set of morphological words of Japanese. Consider the cartesian product, $D_k \times D_j$, of the two sets. A mapping between the sets may be defined as follows.

$$I(D_K) = D_J.$$

implying that the image of Dk is Dj; taking the inverse mapping

$$I^{-1}(D_j) = D_k$$

By generalizing the relation and the mapping between the two sets, we may consider the word set of the source language to be the domain, and the target language word set to be the range. Assuming the same cardinality for both the domain and the range, D_k and D_j may be partitioned as shown below. Here we suppose

$$(k_1, k_2, k_3, \ldots k_n) \in D_k \qquad (j_1, j_2, j_3, \ldots\ldots j_n) \in D_j$$

(a) one-to-one

(b) one-to-many

(c) many-to-many

Obviously, one-to-one correspondence is isomorphic. Thus our attention will be focused on one-to-many and many-to-many relations. The translation of these relations depends on various factors: allomorphs, synonyms and homonyms of both languages. As an elementary strategy for the translation of those correspondence, we adopted a normalization pro-

cedure which ensures the decomposition of one-to-many and many-to-many correspondence into one-to-one correspondence. As for the translation which is dependent on synonyms or homonyms, we specify the canonical form and the semantic feature, respectively. In reality, there are some linguistic representation (words) which exists in Korean but do not exist in Japanese(and the converse in also true); therefore, the need to make new words.

4.2. Word order in the segmented words

Between Korean and Japanese, some common properties are observed, such as an agglutinative language structure and same word order (SOV)[5]. In this subsection, we examine the word order in a segmented word. There are some corresponding properties in word order of the segmented words between both languages as follows:

[property 1] correspondency.

[property 2] inversion.

[property 3] abbreviation.

Among the properties, property 3 depends upon Korean pragmatic information.

Korean and Japanese have a remarkable characteristics; namely, the structure of segmented words. The segmented words are the important language structure as a utterance unit, and play an important role in the analysis of both languages.

The production form of the segmented words can be described in the forms of a regular grammar:

$S \text{ -- > } uB \quad S, B \in N$ (nonterminal symbols)

$S \text{ -- > } u \quad u \in T$ (terminal symbols)

They are both right linear. Denoting the language defined by such a regular grammar by $L = L(G)$ lead to the existence of a finite state automaton M such that $L(G) = T(M) = \{ w \mid M \text{ accepts } w\}$. And, if $L(G) = L(G')$, there is a sequence equivalence such as $S(G) = S(G')$. In other words, for each symbol a in the vocabulary of some regular set R, let R_b be a particular regular set. Suppose that we replace each word $a_1, a_2, a_3, \ldots a_n$ in R by the set of words of form $w_1, w_2, w_3, \ldots w_n$, where w_i is an arbitrary word in Rb. Then the result is always a regular set. More formally, a substitution f is a mapping which is from vocabulary A to subsets (language family) of vocabulary B. Thus the mapping f is extended to strings as follows:

1) $f(\xi) = \xi$, 2) $f(xa) = f(x)f(a)$.

The mapping f is extended to languages by defining

$$f(\mathrm{L}) = \bigcup_{x \in L} f(x).$$

A type of substitution that is of special interest is a *homomorphism.* A homomorphism h is a substitution such that $h(a)$ contains a single string for each a. We generally take $h(a)$ to be the string itself, rather than the set containing that string. It is useful to define the *inverse homomorphic image* of a language L to be

$h^{-1}(L) = \{x | h(x) \text{ is in } L\}$

We also use, for string *w;*

$h^{-1}(w) = \{x | h(x) \text{ is in } w\}$

Consequently, the translation between Korean and Japanese is closed in the substitution among the constituent which are called the segmented words.

4.3. Syntax

It is seen intuitively from the correspondence in the segmented words and word order, that Korean and Japanese have the similar language structure [6]. Let us compare the two parse trees of the actual example sentences.

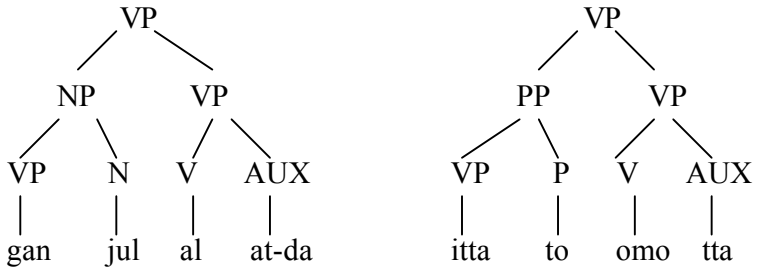


Fig 1: syntactic trees of "(I) thought (somebody) went (somewhere)"

It is obvious that the parse trees correspond to each other in a one-to-one fashion, but the lexical categories do not coincide with each other. This means that both languages do not generate the same set of sentential forms: $S(G) = \{w \in (NUT)^+ \mid S \text{ -*-> } w\}$. Furthermore, there is no algorithm for deciding whether or not two given context-free grammars generate the same sentential forms [9]. This proposition reveals the reason why we adopt the covering grammar and the grammar modification principle.

4.4. Semantics

If a sentence is syntactically ambiguous, it has more than one canonical derivation and is semantically ambiguous if, for a given canonical derivation, it has more than one translation. Derivations are not related directly to a language but to a grammar that generates it. In the translation between Korean and Japanese, there exist several kinds of inherently ambiguous sentences which are generated only by ambiguous grammar of both languages.

In the NARA system, the semantic knowledge is used to eliminate the ambiguity in the syntactic-based translation. But, its role is a minimum essential in the NARA system. Because a semantic theory of natural language, for example situation semantic theory, being underdeveloped, and is not necessary and sufficient condition for the Korean and Japanese translation system. However, for the word that involves the ambiguity in the translation processing, we specify the lexical semantic features and introduce the individual semantic features into the syntactic feature system. In consequence, the lexical semantic features of the constituents are kept in the phrase structure and are applied to the semantic-based translation. That is, the constraints for the semantic sensitive translation are described in the partial phrase structure, and play a role of adjusting semantic sensitive translation.

4.5. K-J Grammar

In this section, we design a K-J (or J-K) grammar which eliminates syntactic or semantic ambiguity of both languages. This grammar corresponds to the communicative competence model for the translation system between Korean and Japanese. The grammar is motivated by grammar modification and covering grammar; the original grammar is not often suitable for a particular parsing technique but can be modified into an equivalent form which is suitable.

ALGORITHM: irregularity categories removal or adjustment and semantic features insertion

Input: a 5 tuple phrase structure grammar $G = (N, T_k, T_j, P, S)$ for the translation.

Output: an equivalent 5 tuple phrase structure grammar $G' = (N', T_{k[sem\text{-}k]}, T_j, P',$ S).

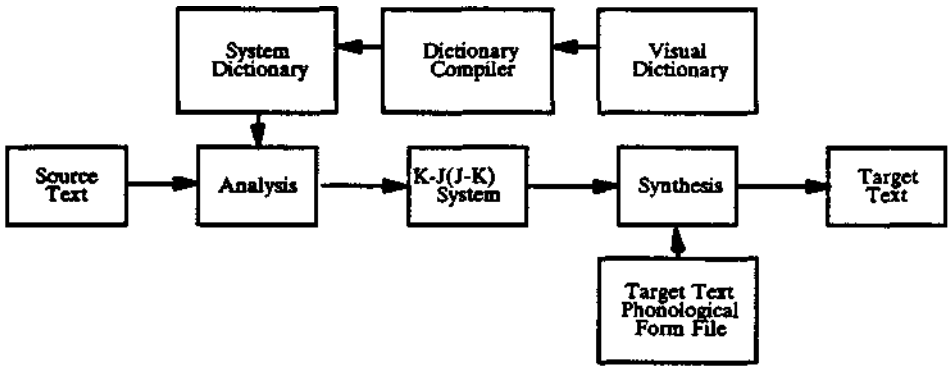
Method: empirical and heuristic method.

Where N and N' are nonterminals, T_k, T_i, T_k, and T_j, are terminals, *sem-k* is semantic features, P and P' are production rules, and S is the start symbol. The J-K grammar is designed by the method analogous to that of the K-J grammar. In the unification-based grammar framework, the semantic features are accepted by a special phrase structure rule, a linking rule (unification), which causes the relevant information about the phrase to be passed down the tree as a feature on the syntactic nodes. Therefore, translation procedure is constructed by a succinct algorithm founded on the K-J(J-K) grammar.

5. The performance model of the NARA system

5.1. System architecture of the NARA system

Before describing the performance model of the NARA system, we briefly describe the NARA system below:



5.2. Analysis

- Morphological analysis

As the preprocessing for the morphological level translation, segmented word analysis is carried out on each word given by the lexicon information.

- Syntactic analysis

The structure of a sentence is analyzed on the phrase structure level. A tree structure which serves as an intermediate structure for translation is constructed.

5.3. K-J(J-K) system

We formulate the internal interface for the translation. This interface corresponds to the transducer of translation. We can define the K-J(J-K) system as a 3-tuple grammar $G=(w_j, w_k, k(\text{or } j))$, where w_k and w_i are Korean words and Japanese words, respectively, $k(j)$: $w_j \rightarrow w_k$ ($w_k \rightarrow w_j$) is the homomorphism. The K-J(J-K) system ul G defines the following sequence preserving the word order:

$$w_k^1 = k(w_j^1), \quad w_k^1 \; w_k^2 = k(w_j^1)\, k(w_j^2), \ldots.$$

It also defines the following language

$$L(G) = \{k^i(w_j) \mid i > 0\}.$$

As mentioned above, the K-J system constitutes a simple device for the translation. A language defined by the K-J(J-K) system corresponds to the target language. Inversely, the mapping j of w_k into w_j is such that the inverse homomorphism

$$j(w_j) = \{w_j \mid k(w_j) = w_k\},\ j = k^{-1}$$

exists. Thus, we define the two-way simultaneous translation system NARA by:

$$j(L_k) = k^{-1}(L_k) = \{w_j \mid k(w_j) \in L_k\}.$$

We can define the NARA system using the extended notion; the inverse homomorphism can be replaced by the direct operation of a finite substitution as follows. Consider a grammar (e.g. Korean) $G_j = (N_j, T_j, P_j, S_j,)$ and let j be a finite substitution, defined on the vocabulary $(N_k \cup T_k)^*$ such that $j(a)$ is a finite (possibly empty) set of word for each word a. We denote

$$j(N_k) = Nj, \qquad j(T_k) = T_j, \qquad P_j \supset j(P_k), \qquad S_j \supset j(S_k).$$

Then, the grammar (e.g. Japanese)

$$G_j = (N_j, T_j, P_j, S_j)$$

is the translation of G_k. If $I(G_k)$, $I(G_j)$ are the sets of all translation of G_k and G_i, respectively, then $I(G_k) = I(G_j)$, and I is an invariant for G_k and G_j.

5.4. Synthesis

From the morphology dependent on the target language is generated with the aid the phonological form file, the correct phonological form of the target language, which is subsequently output.

5.5. Dictionary

The dictionary consists of the K-J(J-K) grammar and the lexicon. A dictionary

compiler is used to transform a visual dictionary into a system dictionary implemented in the form of a B-tree. The modularity of the grammar and the ease way of operation which to update the dictionary serve as major factor in the system.

5.6. Complexity of system NARA

In this section, we present how we can predict the time or memory space or sequential operation that will be needed to perform the computing model of the NARA system, and how the translation process can be specified clearly and unambiguously. The complexity of the algorithm in usually measured by the growth rate of its time or space requirements, as a function of the size of the input (or the length of input string) to which the algorithm is applied. We shall now define the time characteristics of translation process. There are some kinds of syntactical relations such that structural distance is naturally involved in the simultaneous translation. Consider the following example:

[[[tomotachi]ni] [[[kino] [[hisashiburi]ni]] atta]].

This sentence can be translated into Korean as follows:

1) [chingu *ege* [[oje orenman e] mannatta]].

2) [chingu *lul* [[oje orenman e] mannatta]].

Such ambiguity arises in translation due to one-to-many relation on morphological level, the sentence 2) is the well formed translation. The reason is co-occurrence relation; namely, a Japanese verb *a-u (meet)* co-occurs with a postposition *ni* (dative case), and a Korean verb *mana-da* co-occurs with a postposition *lul* (accusative case). If the postposition proceeds the verb, then simultaneous translation is impossible. In this case, delay time $p > 1$ for complete translation is required before two words bind, and one more operation is required to rescan the translated sentence. We refer to this case as quasi-real time translation. We formalize the time complexity of translation. An utterance string of the source language L is the sequence string $S_t\ (L)$. $S_t\ (L) = (k_1, k_2 \ldots k_t)$ is a partial utterance string up to time t, and $K\text{-}J(S_t\ (t))$ is a translation sequence string up to time t. Also $T_t\ (L) = (j_1, j_2, \ldots j_t)$ is a target language which is generated by the K-J system. The translation operates in real time so that delay time is 0. Therefore, $K\text{-}J(S_t\ (L)) = (T_t\ (L))$ where $S_t = T_t$. The translation operates in the quasi-real time so that delay time $p > 1$. Therefore, $K\text{-}J(S_t\ (L)) = (T_t\ (L))$ where $T_t - S_t > 1$. However, the nature of on-line translation is unchangeable.

We compare our translation results to Rene Thorn's hypothesis: the principle of isomorphism concerning linguistic universality. Let T_1 be a text of language L_1, and T_2 be a text to be translated from T_1 into language L_2. Suppose $\{Q_1^i\}$ and $\{Q_2^j\}$ are phrase elements of decompositions of T_1 and T_2, respectively, then the following principle of isomorphism holds:

[Principle of Isomorphism] A one to one correspondence exists between $\{Q_1^i\}$ and $\{Q_2^j\}$ which conserve each signification. Moreover, this correspondence nearly preserves the order of phrase elements; in other words, if the *ith* element Q_1^i of T_1 corresponds to the *jth* element Q_2^j of T_2, then $|j-i| < 4$.

We consider that Thom's hypothesis provides the index for the measurement of the translation complexity between some two languages.

6. Concluding remarks

Our approach for constructing the NARA system included logical study and experimental study; the former was given by the mathematical formalization, the latter by the correspondence of two languages. In the view of computational linguistics, we separated the mechanism of two way simultaneous translation system into the levels of abstract theory, algorithm, and implementation to carve out the results at each level in more independent fashion. In order to do so, we specified four important levels of the description: the lowest level is the morphology, the second level is the segmented words, the third levels are the syntax and the semantic, and the top level controls the computing model of each level. Hence, we could determine the range of correspondence between internal representations of both grammars, and the basic architecture of the machinery actually instantiates the algorithm. Consequently, our model produces the extra power by the proposed theory with multiple levels of representation and systematic mapping between the corresponding levels of two languages, because translation efficiency requires both a functional and a mathematical argument. In the view of software engineering, going through each level of abstraction we expect to make an elegant program which satisfies the requirements of the machine translation system such as simplicity, reliability, adoptability and modularity. Nevertheless, the complete pragmatic translation remains quite obscure.

Acknowledgement

The author is deeply grateful to Dr. Gil Rok Oh and Dr. Min Ho Kang for their encouragement.

References

[1] Arden, B., ed, *What can be Automated?,* M.I.T. Press, Reading, 1980.

[2] Berwick, R., and Weiberg, A., *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition,* M.I.T. press, Reading, 1984.

[3] Berwick, R., *The Acquisition of Syntactic Knowledge,* M.I.T. press, Reading, 1985.

[4] Chomsky, N., *Aspects of the Theory of Syntax,* M.I.T. Press, Reading, 1963.

[5] Chung, H., *Current Korean: Elementary Sentence Patterns and Structures,* Komasholin, Reading, 1982 (in Japanese).

[6] Chung, H., *Korean language Information Processing,* Ph.D. dissertation, Tokyo Univ., 1986.

[7] Chung, H and Kunii, T., *A Two-way Simultaneous Interpretation System between Korean and Japanese: NARA,* Proceeding of COLING'86, 1986.

[8] Culik, L. and Salomaa, A., *On the decidability of homomorphism equivalence for language,* Journal of Computer and System Science, 18, 1978.

[9] Harrison, M., *Introduction to Formal Language Theory,* Reading, Addison- Wesley, 1978.

[10] Niholt, A., *Context-free Grammar: Cover, Normal Forms and Parsing,* Springer, Reading, 1980.

[11] Salomaa, A., *Jewels of Formal Language Theory,* Computer Science Press, Reading, 1981.

[12] Thom, R., *Topologie et Linguistique,* in Essays on Topology and Related Topics, Springer, 1970.