Knowledge-Based Machine Translation

Sergei NIRENBURG

ABSTRACT: This paper provides an overview of the KBMT-89 project at Carnegie Mellon University's Center for Machine Translation, as well therefore of the special number of this journal, which reports on the project. The knowledge-based approach to machine translation is presented and defended in a historical context. Various components of the system, key parts of which are described in subsequent papers of the issue, are introduced and paired with their computational motivations.

KEYWORDS: English, interlingua, Japanese, knowledge-based machine translation

1. INTRODUCTION

Historically, machine translation systems have been of three major types: direct, transfer and interlingua. Detailed descriptions of the three approaches, with all their modifications and varieties, can be found in the machine translation literature (see in particular Hutchins, 1986; and Zarechnak, 1979). We will comment only briefly on the essential differences.

The direct translation systems typically rely on a large set of language-pair-dependent rules to carry out the translation of a text. These rules take separate grammatical and lexical phenomena of the source language (SL) and their realizations in the target language (TL) and put the two in correspondence. Examples of such systems are SYSTRAN (Wheeler, 1984) and PAHO (e.g., Vasconcellos and Leon, 1985). A schematic representation of processing in a direct translation system is given in Figure 1.

![Figure 1. A direct translation system.](image)

Transfer systems involve a measure of target-language-independent analysis of the source language. This analysis is usually syntactic. It allows substituting

*Machine Translation* 4 (1989), 5 -- 24
source language lexical units with target language lexical units in context. That is, it permits taking into account the types of syntactic sentence constituents in which lexical units appear. Among the transfer translation systems are EUROTRA (Arnold and des Tombe, 1987) and METAL (Bennett, 1982). A transfer translation system is schematicized in Figure 2.

![Figure 2. A transfer translation system.](image)

In interlingua systems the source language and the target language are never in direct contact. The processing in such systems has traditionally been understood to involve two major stages: representing the meaning of a source language text in a language-independent formal language, interlingua, and then expressing this meaning using the lexical units and syntactic constructions of the target language. Few interlingua systems have been fully implemented because of the very significant complexity (both theoretical and empirical) of extracting a “deep” meaning from a natural language text. Among the systems conforming to the interlingua design are CFTA (Vaucouleurs, 1975) and TRANSLATOR (Nirenburg et al., 1987). A schematic view of processing in an interlingua translation system is seen in Figure 3.

![Figure 3. An interlingua translation system.](image)

for producing an adequate meaning of a source language text.

As a rule, however, machine translation researchers who believe in translating without “deep” understanding (or perhaps who believe in the unattainability of “deep” understanding) of the source language text tend to prefer the transfer paradigm. The price they have to pay for avoiding meaning analysis is the need for an extra step in the translation process, namely, postediting.1

Builders of interlingua systems, on the other hand, must devise new knowledge-based natural language understanding techniques to account for some of the more difficult problems in natural language processing. The complexity of this task induces knowledge-based machine translation researchers to constrain the range of phenomena processed by their systems. This is usually done by restricting the sublanguage of translation to a relatively small subset of a natural language.2

The first forty years of MT experience teach us that significant long-term progress depends on advances in our basic knowledge of how to model natural language understanding using computers. It may be argued that systems with limited understanding might be immediately more useful and practical,3 but systems that extend and strengthen our grasp of the problems of translation must strive for understanding. This will be true even if, at the beginning, the lack of knowledge about specific types of disambiguation and meaning representation require the introduction of a measure of human involvement in the form of

---

1The relative functions and purposes of pre-editing, post-editing and interactive editing are for the most part intuitively clear. For a more detailed discussion, see several of the papers in Nirenburg, 1987.

2It is not possible to include a full-fledged discussion of the relative merits and faults of each of the approaches to MT. Additional information and considerations with respect to the several machine translation paradigms can be found in numerous publications in the field; see, for instance, Nirenburg, 1987; Nirenburg et al. 1989; and Hutchins, 1986.

3These claims, however, are not universally convincing either.
interactive editing.

2. SPECIFICATIONS AND ARCHITECTURE

The KBMT-89 project is devoted to creating a working prototype of a machine translation system with the following specifications:

- Source languages: English and Japanese;
- Target languages: English and Japanese;
- Translation paradigm: Interlingua;
- Computational architecture: A distributed, coarsely parallel system; and
- Subworld (domain) of translation: personal computer installation and maintenance manuals.

The knowledge acquired for the system includes:

- An ontology (domain model) of about 1,500 concepts;
- Analysis lexicons: about 800 lexical units of Japanese and about 900 units of English;
- Generation lexicons: about 800 lexical units of Japanese and about 900 units of English;
- Analysis grammars for English and Japanese;
- Generation grammars for English and Japanese; and
- Specialized syntax $\Rightarrow$ semantics structural mapping rules.

The underlying formalisms that were developed for the use in this system are:

- The knowledge representation system FRAMEKit;
- A language for representing domain models (a semantic extension of FRAMEKit);
- Specialized grammar formalisms, based on Lexical-Functional Grammar;
- A specially constructed language for representing text meanings (the interlingua); and
- The languages of analysis and generation lexicon entries, and of the structural mapping rules.

The procedural components of the system include:

- A syntactic parser with a semantic constraint interpreter;
- A semantic mapper for treating additional types of semantic constraints;
- An interactive augmentor for treating residual ambiguities;

Figure 4. Architecture of the KBMT-89 system. ('SL' and 'TL' designate 'source language' and 'target language'; 'ILT' stands for 'interlingua text'; 'F-s' represents 'f-structure'; 'E' and 'J' designate 'English' and 'Japanese'; the languages used in KBMT-89; and 'MRs' stands for 'mapping rules.')
A semantic generator producing syntactic structures of the target language, complete with lexical insertion; and

A syntactic generator, producing output strings based on the output of the semantic generator.

The support and environment facilities in KBMT-89 include:

- A knowledge acquisition tool for acquiring ontologies and lexicons, ONTOS;
- A knowledge acquisition tool for acquiring grammars; and
- Testing environments for analysis, augmentation and generation.

2.1. The KBMT Architecture

KBMT-89 takes as input single sentences of English or Japanese and produces representations of their meanings in a specially devised notation, called interlingua. The representation resulting from analyzing a unit of input is called an interlingua text or ILT. Taking an ILT as input, the generator produces sentences in Japanese or English that are translations of the original input sentences. Figure 4 illustrates the global architecture of the system.

2.1.1. The Analyzer

The analyzer consists of two intimately interconnected components — a syntactic parser and a semantic interpreter, called the ‘mapping rule interpreter.’ The syntactic parser obtains the source language input and produces a syntactic structure for it. The parser uses an LFG-type grammar, so that the resultant syntactic structure is, in fact, an LFG f-structure.

Analysis grammar rules have this form (the following is actually the topmost phrase-structure rule):

```
(<START> <--> (<EXPR <END-FUNC>)
  (X0 = X1)))
```

Following is a sample f-structure:

```
((NUMBER-BULLET ((ROOT 1) (VALUE 1)))
 (OBJ
  (*OR*
   ((CASE ACC) (DUAL NIL) (NEAR NIL) (SPECIFIC NIL) (REF DEFINITE)
    (DET ((ROOT THE) (REF DEFINITE)))
    (ROOT TAPE) (PERSON 3) (NUMBER SINGULAR)
    (COUNT NO) (MEAS-UNIT NO) (PROPER NO)))
   ((CASE ACC) (DUAL NIL) (NEAR NIL) (SPECIFIC NIL) (REF DEFINITE)
    (DET ((ROOT THE) (REF DEFINITE))))
  )))
```

As soon as the f-structure for the source language sentence is created, the semantic interpreter starts applying mapping rules in order to substitute source language lexical units and syntactic constructions with their interlingua translations. Roughly, lexical units map into instances of domain concepts (e.g., the English data will map into the interlingua information), while syntactic structures map into conceptual relations (e.g., subjects of English sentences often map into the agent relations). The process of mapping-rule application is accompanied by elimination of analysis ambiguities through the application of semantic constraints on co-occurrence of various concept instances. Here follows as example the lexical mapping rule for the unit 'joy stick':

```
("joy stick" (CAT N)
 (CONJ-FORM SINGULAR)
 (FEATURES
  (CLASS DEFAULT-NOUN-FEAT)
  (all-features ((PERSON 3) (NUMBER SINGULAR) (COUNT YES)
    (PROPER NO)
    (MEAS-UNIT NO) (ROOT JOY STICK)))
 (MAPPING
  (local ; From the ontology
    (HEAD (JOY-STICK)))
  (CLASS OBJECT-MAP)) ; Inherits structural
 ; mapping rules from
 ; OBJECT-MAP class
```

And here a structural mapping rule for English nouns:

```

4This description is simplified for clarity. In reality, mapping rule application starts as soon as an f-structure is produced for any structure component and not after the entire sentence is processed.
```
(OBJECT-MAP
(mapping (local
(slot
(NUMBER == number)
(REférence == ref)
(PERSON == person)
(CARDINALITY == quantity)
(NEAR == near)
(*PREDEFINED-SLOT* == modifiers)
(*PREDEFINED-SLOT* == pre-nom-noun)
(*PREDEFINED-SLOT* == post-nom)
(*PREDEFINED-SLOT* == post-nom)
(Member == member)
(dual == dual)
(specific == specific)
(near == near)
(intense == intense)
(new == new)
))
)

And here is the result of mapping-rule application to the f-structure shown earlier:

[*REMOVE
(*STICKY-TAPE
(NUMBER SINGULAR)
(REférence DEFINITE)]]

(SOURCE [*DISKETTE-DRIVE
(NUMBER SINGULAR)
(REférence DEFINITE)]]

(TENSE PRESENT)
(MOOD IMPERATIVE)
(NUMBER-BULLET [*ANY-NUMBER
(CARDINALITY 1)]))

The general architecture of the KBMT-89 analyzer is given in Figure 5.

2.1.2. The Knowledge Plane

The meaning of the input text is, as noted above, represented in a specially designed knowledge representation language, an interlingua. In KBMT-89 the interlingua is in turn represented in a frame notation and thus can be viewed as a kind of a semantic network. Like other artificial or formal languages, interlingua has its own lexicon and syntax. Yet while the syntax of the interlingua is independently motivated, its lexicon is based on a model of the domain (or 'world') from which the texts to be translated are taken. In the case of KBMT-89, the text is the domain of personal computer installation and maintenance. Thus, interlingua nouns are object concepts in the ontology; interlingua verbs correspond,

3We sometimes use the terms 'ontology' or 'concept lexicon' to refer to domain models. Cf. Footnote 1 in the paper “Lexicons” by Gates et al. in this issue.

4In KBMT-89 the inputs were restricted to single sentences, and therefore the need for the text-level index did not arise.
hesion (e.g., therefore). The latter are represented in interlingua using special formalisms not connected with the ontology. An ILT is schematicized in Figure 7.

Figure 7. A sample ILT.

Figure 8 illustrates processing as seen from the point of view of data connections among the three lexicons (analysis, generation and concept) and the ILT.

2.1.3. The Generator

The generation component of KBMT-89 takes an ILT as its input and produces a target language text as its output. Our generator consists of two major modules, one semantic and one syntactic. The former, usually referred to as the 'f-structure builder,' performs the tasks of target language lexical selection and choosing among target language syntactic constructions; it is aided in these tasks by the generation lexicon and the generation structural mapping rules, respectively. The output of this module is an f-structure of the target language sentence that will be output by the system. As its syntactic module KBMT-89 uses GENKIT (Tomita and Nyberg, 1988). The KBMT-89 generator is a subset of the DIOGENES generator (Nirenburg et al., 1988a).

The generator is discussed in detail in the paper by Nyberg, McCordell, Gates and Nirenburg in Part II of this issue. At this point, we will simply illustrate the architecture of the generation module (which is in many ways similar to the analyzer architecture), in Figure 9; and the process of lexical selection, in Figure 10.
Figure 8. The interaction among lexicons and ILT. Note that some source language lexical units are connected to their interlingua meanings directly, bypassing the concept lexicon. The figure also illustrates the lack of symmetry in the treatment of lexical semantics in analysis and generation; the main problem in analysis is polysemy, while in generation it is synonymy.

The following is an example of a generation-oriented structural mapping rule:

```
(maprule s
  *PROPOSITION
  :exclusive
  (and (frame-agent @frame.theme
                  (or (eq @frame.clauseid.focus 'agent)
                       (null @frame.clauseid.focus)
                       (lex-feature-p 'valency (lex 'trans)
                                      (not
                                        eq @frame.agent 'UNKNOWN)))))
    ;; agent is defined
    ((slot agent) => (slot sub)))
  ((slot theme) => (slot obj)))
```

And here is an example of a generation grammar rule:
2.1.4. The Augmentor

It should perhaps already be apparent that there is a difference between our illustrations of ILT formats and the output from the parser’s mapping rule interpreter. There are several reasons for this phenomenon. Among the most important are compatibility between the parser’s output structures and the input structures of the generator (that is, the ILTs); constraints on the formulation and applicability of mapping rules in semantic interpretation; and the requirements for representing in interlingua some noncompositional facets of the overall meaning of the sentence, such as speech act and discourse cohesion.

Our augmentor serves two main purposes. First, it reformats the output of the analyzer in the canonical ILT formalism. Second, it helps eliminate residual ambiguities (that is, multiple candidate ILTs for a given input sentence) by applying additional semantic and pragmatic constraints and, if that fails (typically, due to the unavailability of a unit of knowledge), by entering a dialog mode with the users and facilitating their decisions about disambiguation. The architecture of the augmentor is illustrated in Figure 11.

2.1.5. The Tools

A system of KBMT-89’s size and complexity cannot be successfully completed in the absence of a set of tools that supervenes on a) the acquisition of the very large quantities of world and linguistic (lexical and grammatical) knowledge, and b) the process of testing and debugging all the system’s processing components.

The major knowledge acquisition aid in KBMT-89 is ONTOS, a system for interactive acquisition and maintenance of domain models; it is also used for acquiring the analysis and generation lexicons. ONTOS was specifically developed for use in KBMT-89; however, it has an independent value in that it can be applied in a number of situations wherein the acquisition of large quantities of knowledge is required. Figure 12 depicts the screen of an IBM PC RT on which ONTOS is running. The screen displays a small segment of the project’s ontology.

Figure 10. A representation of the lexical selection process.
3. EXTENSIONS

As is the case with practically any software system, each and every module of KBMT-89 can be further improved. The practical impetus to improvements will come from the growth in the size and complexity of the sublanguages that the system will be called upon to process. New syntactic structures, lexical units and types of meanings will have to be included in the system's inventory.

One of the most important extensions is treating intersential text. The representational substrate of KBMT-89 is sufficiently broad to allow an experiment in treating intersential text. The ILT contains a provision for representation of intersential discourse cohesion markers and of flexible-scope focus values. The generator is a subset of the DIOGENES generator (Nirenburg et al., 1988a) that is already able to process intersential anaphora. The automatic augmentor module is also geared to processing intersential anaphora.

Other major extensions will involve introducing additional source and target languages and supporting translations in new domains.

The expressive power of the interlingua will have to be extended to cover more lexical and pragmatic meanings. The existing grammars and lexicons will have to be enhanced both in their coverage and organization. Better treatment of multi-sense lexemes is required, for instance, as well as a re-evaluation of the size of the set of lexical categories used by the grammar. From a practical point of view, special preprocessors are needed to deal with such issues as
treated translatable and untranslatable material in figures and tables; translating abbreviations and special symbols; reproducing the layout characteristics of the input text in the output, etc. Other subsystems must be designed and written; these include a general-purpose morphological processor for the analyzer and the generator and an accompanying acquisition tool for morphological knowledge.

A separate set of improvement-related tasks concerns performance optimization. KBMT-89 has been built as a working prototype of an interlingua system working with a set of large grammars and lexicons and a non-trivial domain model. A number of steps can be taken to improve the performance of the system. Thus, unambiguously grammars can be written; the lexicons can be tuned to the needs of a particular, however large, corpus so that no extra lexical mappings occur and there are no unused concepts in the domain model. At the same time, a number of techniques can be used to speed up the component programs. Thus, for instance, the Lisp code that is automatically generated for the generation grammars may be improved by adding optimizing steps in the compiler. We found it necessary to begin adding optimizing steps during the final stages of the project, when the English generation grammar became very large. Currently, lexical insertion is via rules in the grammar; in the future, this will be supplanted with a separate dictionary and morphological rules so each inflected form of a root noun or verb need not be coded directly in the grammar. This will have a two positive effects: It will decrease the size of the compiled grammar and increase the productivity of the grammar writer.

In addition to its utility as an MT shell, KBMT-89 can be profitably used as a research tool and testbed in computational linguistics and artificial intelligence.

In its current state the system provides an excellent tool for devising and testing new and more powerful specialized semantic interpretation algorithms, such as, for instance, noun-noun compound understanding or prepositional phrase attachment. With more types of semantic and pragmatic knowledge appearing in the ILT, more specialized “microtheories” will be devised and/or incorporated into the process. The modularity of the KBMT-89 architecture will facilitate this. In fact, the system already allows a greater measure of modularity, which has been tested partially in the DIOGENES project and will probably have a blackboard-based architecture underlying it in its next incarnation.

Our generation component is also a good substrate on which to build more sophisticated generators. In particular, it facilitates the interaction of syntactic, lexical and prosodic processing and offers a level of reliance on world knowledge that is unusual in most natural language generators.

An additional advantage of using KBMT-89 as a research vehicle is that it is a comprehensive system that allows immediate testing of a new component in the context where a genuine output can be obtained.

The interface component of KBMT-89 can serve as a medium for building other interfaces, notably for computer-aided instruction and, in particular, for teaching foreign languages. A comprehensive understanding-and-generation system like KBMT-89 can also be used as a component in a cognitive agent model, integrated with components for planning and problem solving, perception and action simulation. It can be useful in machine learning systems, especially those studying learning from text or learning by being told, or in systems that investigate hybrid learning processes.

The ontological and domain knowledge facet of KBMT-89 can be quite useful for research in the area of acquisition and maintenance of very large, multifunctional knowledge bases. Even in its current state, ONTOS is used in over half a dozen different projects at Carnegie Mellon University and elsewhere.

The computing technology embodied by KBMT-89 has a number of potential practical applications outside machine translation proper. One of the areas in which KBMT-89 can yield immediate practical results is in design and development of high-quality translator’s workstations. The interaction environment can be extended to include other types of human-computer interaction. Additional knowledge sources (for instance, human-readable dictionaries and encyclopedias) can be connected to the system. And the presence of the analyzer and the generator will allow the system to suggest solutions to the human translator; this is a feature that all current translator’s workstations necessarily lack.

The technology developed in KBMT-89 is readily usable in applications that require different types of inputs and/or outputs to a natural language processor. Thus, instead of forwarding an interlingua text to the generator, one could pass it on to a special reasoning program to produce an abstract of the input text, answer questions based on it or place the input text into one of a number of taxonomic classes. KBMT-89 can be also reconfigured for supporting natural language interfaces to database systems. Indeed, if a data manipulation (query) language is substituted for the interlingua, the task of query formulation can become quite similar to that of analyzing a natural language input for translation.

It is appropriate at this point to extend thanks to several people who contributed to KBMT-89: Jaume Carbonell, Director of the Center for Machine Translation, for advice on design and for participation in many key discussions; Masaru Tomita for maintaining and improving the parser when the project needed it; and Radha Rao and Annamarie Mackulak, for extraordinary administrative and secretarial support.
REFERENCES


