# The Universal Parser Architecture for Knowledge-Based Machine Translation

Masaru Tomita and Jaime G. Carbonall[1]
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213

## Abstract

Machine translation should be semanticalty-accurate, linguistically-principled, user-interactive, and extensible to multiple languages and domains. This paper presents the *universal parser* architecture that strives to meet these objectives. In essence, linguistic knowledge bases (syntactic, semantic, lexical, pragmatic), encoded in theoretically-motivated formalisms such as lexical-functional grammars, are unified and precompiled into fast run-time grammars for parsing and generation. Thus, the universal *parser* provides principled run-time integration of syntax and semantics, while preserving the generality of domain-independent syntactic grammars, and language-independent domain knowledge bases; the optimized cross product is generated automatically in the precornpllation phase. Initial results for bi-directional English-Japanese translation show considerable promise both in terms of demonstrating the theoretical feasibillty of the approach and in terms of subsequent practical utility.

## 1. Introduction

Accurate translation requires a degree of comprehension, and several projects have developed prototype systems to demonstrate the feasibility of knowledge-based machine translation [2,12,11]. These approaches combine syntactic and semantic information to produce an intermediate knowledge representation of the source text[2] which is then generated in the target language. This paper does not attempt to revisit the ample rationale for the knowledge-based machine translation concept - such discussion may be found in the literature [4.2,12] - but rather discusses the *universal parser* (UP) approach for combining syntactic, semantic and lexical knowledge in order to analyze source text for re-generation in multiple target languages, preserving imvariance of meaning.

First, however, let us review the set of performance objectives that contributed to the design of the universal parser:

- *Semantic accuracy* - The translation should maintain *semantic invariance* above an else. Paralleling syntactic form, maintaining equivalent length of text, and other such criteria are considered of secondary importance. Thus, the knowledge-based approach was the only logical choice.
- Multi- II*ngual generality* - The system should be able to handle any natural language and any semantic domain. The addition of any new language should enable immediate translation to and from all the previous languages, without requiring explicit hand-built transfer grammars for all pairs of languages.
- *Interactive Translation* - Translation should occur in real time, interacting with the user if required. Whereas most existing practical machine translation systems are designed to batch-process large documents, there is a growing need for Immediate interactive translation of business letters, telexes, and eventually interactive communication (by telephone, at an airport counter, at a foreign hospital, etc.). Such interactive usage adds the following demands:
  - *No post-editing* should be required, as one cannot cany along a personal post-editor in case he or she is needed.
  - *Real-time performance* is an absolute requirement, as participants in a dialog will not wait minutes or hours for a response.
  - *Speech compatibility* is an equally strong requirement, as the utility for real-time KBMT translation systems increases dramatically when coupled with speech recognition and synthesis. Speech recognition imposes the requirement to handle unsegmented input, with multiple word candidates present at any point in the input stream. (I.e., the input is typically a lattice rather than a linear string [8].)
- *Linguistic Generality* - Linguistic information (syntactic, semantic, and lexical) should be expressed in elegant, theoreticalty-motivated formalisms - ones that linguists can use to develop and modify grammars and knowledge bases rapidly (such as LFG).
- *Dlscourse Phenomena* - Extra-sentential phenomena such as anaphora, ellipsis, metalanguage, and speech acts, should be handled within the framework, as should inference required to support cross-linguistic variation (such as politeness levels, inference of missing constituents - e.g.. subjects in Japanese - and finer grain lexical selection required in some target languages).
- *Multiple Utility* - In addition to machine translation proper, the methods developed should be applicable to multi-lingual natural language Interfaces (to data bases, expert systems, etc.). and other application such as automated skimming and indexing of texts.

We have achieved the majority of these objectives In an experimental system at CMU's center for machine translation, and we are actively working on developing the other capabilities. The system, consisting of the *universal parser and universal generator,* is an open-architecture approach to knowledge-based machine translation, integrating multiple off-line knowledge sources into a fast on-line run-time system [16]. We have chosen English and Japanese as our initial languages, and simple doctor-patient communications as our initial test domain, and have produced real-time, semantically-accurate, bi-directional translations at the sentential level. The rest of this paper gives an overview of the parser architecture and our system. For more detail, see [17].

[1]Many other members of CMU Center for Machine Translation have made contributions, directly or indirectly. Teruko Watanabe has been developing the Japanese LFG grammar, as well as helping many other parts of the project. Donna Gates has been developing the LFG grammar for English. Marion Kee has developed the semantic knowledge for the domain of doctor/patient communication. Lori Levin Peggy Anderson and Yuko Tomita also contributed in grammar development and/o semantic knowledge development. Kazuhiro Toyoshima has implemented the run time parser. Hideto Kagamida has implemented the syn/sem compiler, the LFG compiler and the LR table compiler. Eric Nyberg, Phil Franklin, Ron Grider and Satoshi Tojo have developed sentence generators for English and Japanese. Hiroaki Saito helped to por and maintain the entire system. Other members who made indirect contributions in many ways are Ralph Brown, Mike Calvin, Alex Hauptmann, Roland Hausser, Fumi Katau, Kevin Knight, Steve Morrieson, Hiroaki Mushu, Sergei Nirenburg, Radha Rao and Michael Witbrock.
Funding for this project is provided by several private institutions and governments agencies in the United States and Japan.

[2]This intermediate semantic representation is often call the *interlingua*, though that is a misnomer. The semantic representation is encoded in a completely formal, canonica and unambiguous notation such as first-order logic or frame-based representations Once the meaning representation is extracted it may be re-generated in multiple languages, paraphrased, summarized, stored, fleshed out by an inference procedure, o otherwise processed.

## 2. The Universal Parser: a New Architecture for Multi-lingual Parsing

Multi-lingual systems require parsing multiple source languages, and thus a universal parser, which can take a language grammar as input (rather than building the grammar into the interpreter proper) is much preferred for reasons of extensibility and generality. However, semantic information can remain invariant across languages (though, of course, not across domains). Therefore, it is crucial to keep semantic knowledae sources separate from syntactic ones. If new linguistic information is added it will apply across all semantic domains, and if new semantic information is added it will apply to all relevant languages. The question, of course, is how to accomplish this factoring, without making major concessions to either run-time efficiency or semantic accuracy.

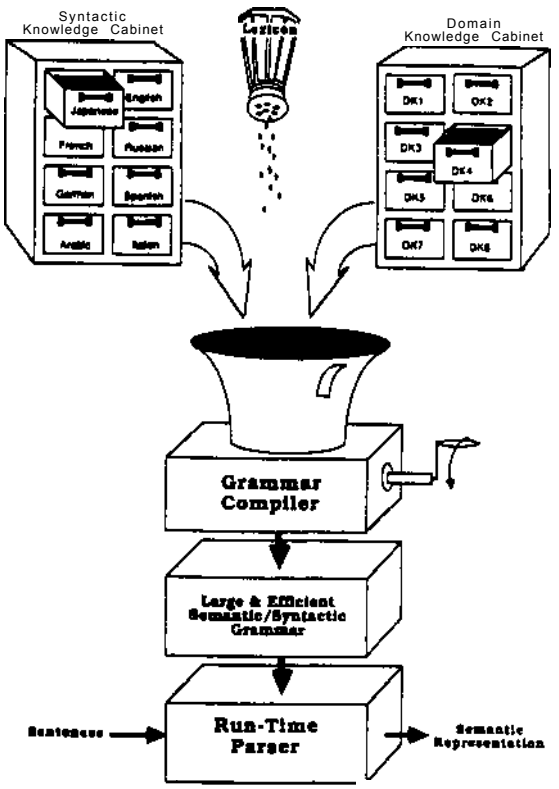


Figure 2-1: Universal Parser Concept

The idea of the Universal Parser is depicted in figure 2-1. There are two kinds of knowledge sources: one containing syntactic grammars for different languages and the other containing semantic knowledge bases for different domains. Syntactic grammars and domain knowledge bases are written in a highly abstract, human-readable manner. This organization makes them easy to extend or modify. The grammar compiler takes one of the syntactic grammars (say Language $L_i$) and one of the domain knowledge bases (say Domain Dj), along with mapping rules (that determine which semantic concept is expressed by what word and what linguistic structure), and produces an object grammar, containing an optimized legal cross-product of both syntactic and semantic information. Whereas the compiled grammar is not human-readable, It must is extremely machine-enicient in terms of on-line run-time parsing speed. When the user inputs sentences in Language L, (and Domain D,), the run-time parser parses the sentences very efficiently, referencing only the compiled grammar, and producing semantic representations of the sentences.

## 3. The System Architecture

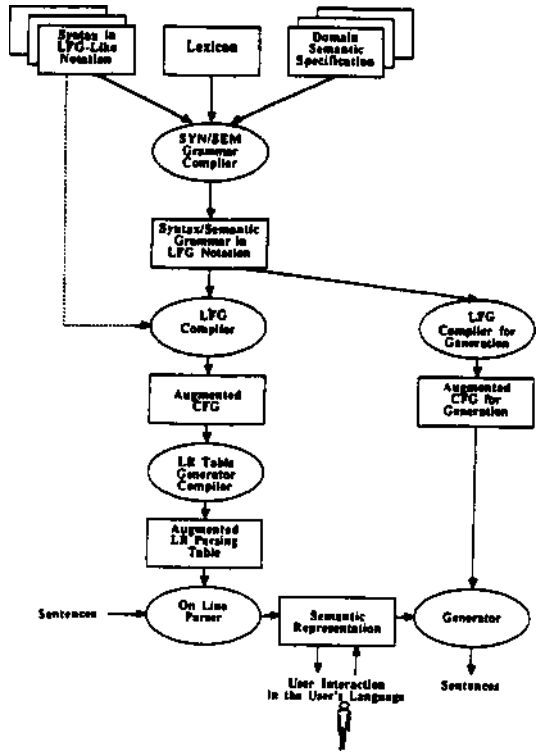


Figure 3-1: System Architecture

Figure 3-1 shows the architecture of the universal parser. We adopt *semantic case frames* for domain knowledge representation and the *functional grammar formalism* for syntactic grammar representation. The run-time grammar produced by the multi-phase compiler is an *augmented context-free grammar (ACFG)* which is further compiled into an *augmented LR table* to be used by a run-time parser based on the Tomita parsing algorithm. These components are described in detail in the following subsections.

### 3.1. Semantic Frame Representation

```
("ACTION
    (ia-a (valua *SENTENTIAL))
      whan (sem *TIME))
      •tart (sem "TIME))
    (:and (SEM "TIME))
    (:fxaq (aaa *FREQUENCY))
    (:duration (sem "DURATION)) )

("PATIENT-ACTION
    (is.-a (valua "ACTION))
    (:agant (earn *PATIENT)) )

("INGEST-MEDICINE
    (ia-a (valua *PATIENT-ACTION))
    (:object (.am *MEDICINE))
    (:inga.t-with (sem *FOOD "MEDICINE)) )

(*MEDICINE
    (ia-a (valua *PHYSICAL-OBJECT))
    (:quant (sem *MEDICINE-QUANTITY)) )
```

Figure 3-2: Fragment of Domain Semantics Specification

We use FrameKit [3] as our knowledge representation language to encode domain semantic knowledge. FrameKit is a compact and fairly efficient general-purpose frame-representation language with multiple-Inheritance, procedural attachment, and default semantics. Domain knowledge consists of a set of frames organized into an inheritance hierarchy. Each frame represents a concept such as object, event, etc. in the domain with appropriate semantic links to other

related frames in the hierarchy. Frames encode typing information, functional dependencies and express compositional constraints used in the parser to block non-productive computations.

Let us consider the domain of simple doctor-patient conversations. Entities in this domain include an event frame *INGEST-MEDICINE and object frames 'MEDICINE,'PATIENT and so on. Example frame definitions are shown in figure 3-2. Sentences with different surface forms that should be recognized as instantiations of these frames include the following examples.

Take the medicine with three glasses of water
every six hours for two days.
What medicine did you take today?
I took two aspirin three hours ago.

As a more direct example, the final semantic representation of the sentence Take the medicine with three classes of water every six hours for two days* produced by instantiatinp, frames is shown in the first sample translation in the appendix. This knowledge structure may then be given to any back-end process, whether it be a language generator (to translate into the target language), a paraphraser. a data-base query system, or an expert system.

### 3.2. The Functional Grammar Formalism

We adopt the functional grammar formalism for syntactic knowledge representation of each particular language. Two wel-known functional grammar formalisms are Functional Unification Grammar (UG) [9] and Lexical Function Grammar (LFG) [1]. Figure 3-3 is a LFG grammar fragment written In a notation similar to PATR-ii [13].

```
(<DEC>  <==>  (<NP> <VP>)
        (((x1 case) = nom)
         ((x2 form) =c finite)
         ("OR"
          (((x2 :time) = present)
           ((x1 agr) = (x2 agr)))
          (((x2 :time) = past)))
         ((x0) = (x2))
         ((x0 :mood) = dec)
         ((x0 subj) = (x1))))
```

Figure 3-3: Fragment of English LFG in the PATR-like Notation

There are two main advantages of using the functional grammar formalism in multi-lingual NLP systems over more traditional linguistic theories:

- A grammar in this formalism can be used for both parsing and generation. Thus, we do not need to write separate grammars tor parsing and generation.
- Functional grammar formalisms such as UG and LFG are well-known among oomputational linguists, and therefore they need not be trained (with some Justifiable resistance) to write grammars in arcane system-specific formalisms.

### 3.3. Grammar Compilation and Efficient On-Line Parting

The previous two sections have described how to represent domain semantics and language syntax. The universal parser unifies both knowledge sources and optimizes the grammar for runtime performance in a series of offline Precompilation phases. The first compear named *syn/sem grammar compter* compiles the syntactic and semantic knowledge, as weft as morphological rules and dictonary, into a single large LFG grammar called *syn/sem grammar.* The compiled syn/sem grammar is basically the aame as its original syntactic grammar except that it acquired many additional semantic equations generated automaticaly by the compiler. The semantic aquations check samantic constraints and bulld samantic representations rather than syntactic f-structures.

This syn/sem grammar in PATR-like notation is further oomplled into an *augmented context-free grammar* (ACFG) by the second oompiler named the *LFG compiler,* this ACFG grammar is represented by a set of context-free phrase structure rules, each of which is augmented with a Lisp program for *test* and *action* as In ATNs. AM the Lisp functions are generated automatically by the oompiler from the constraint equations in the syn/sem grammar. Also note that those Lisp functions are further compiled down to machine code by the standard LISP compiler.

Once we have a grammar in this form, we can apply efficient context-free parsing algorithms. In fact, we subject this grammar to a final round of completion, where the context free rules are compiled into a large augmented LR table for a generalized shift-reduce *parser* based on the Tomita algorithm [14,15]. Whenever the parser reduces constituents into a higher-level nonterminal using a phrase structure rule, the Lisp program associated with the rule is simply evaluated. The Lisp program handles such tasks as:

- blocking partial parses that violate syntactic or semantic constraints (thus enforcing subject-verb agreement, type checking on the arguments to a proposed semantic relation, etc),
- constructing a semantic representation of the input sentence from Its constituent parts (an instantiated frame or causally related set of frames), and
- passing attribute values among constituents at different levels In order to have the information that is needed to perform the constraint-checking and frame-instantiation tasks.

The Tomita algorithm has three major advantages for real-time parsing over other methods:

- The algorithm is fast, due to the LR table precompilation.
- The efficiency of the algorithm is not affected *very* much by the size of its grammar, once the LR parsing table has been precompiled.
- The algorithm parses a sentence *on-line,* i.e., strictly from left to riant and it starts the moment the user types the first word, without waiting for completion of the sentence.

## 4. Concluding Remarks

The first pilot integrated implementation of the universal parser was completed and demonstrated in October 1986. demonstrating the computational feasibility of the concept.

We have written a fairly comprehensive English syntactic grammar and Japanese syntactic grammar In LFG, each containing somewhat under 1000 rules of grammar and regular morphology, The English grammar handies declaratives. Imperatives, yes-no questions, "wh"-questions and other gapped constructions, auxiliary complexes and related phenomena. Additionally we built grammar rules for specialized constructions such as times and dates. The Japanese grammar corresponds roughly in coverage to the English grammar, in addition to having far more comprehensive morphological analysis rules (in LFG notation) required for Japanese. Both grammars are still being refined and extended to achieve full syntactic and morphological coverage. We have started grammar development for a third language, French, to make our system tri-lIngual.

We also developed a non-trivial domain semantic knowledge base in FrameK it for certain classes of doctor-patient conversations, plus mapping rules and a corresponding lexicon including over 500 disease names. This domain was chosen as our test bed for developing the universal parser and generator architectures, and we are currently starting on a second domain (small computer manuals).

All modules are programmed in Common Lisp and running on Symbolics 3600s, HP Bobcats, and IBM RTs - the entire system should be portable to any other workstations running Common Lisp (Explorer, Micro Vax, Sun, etc.)

One of our main research activities at present lies in the area of discourse, as our Initial system operates only on a sentential basis. First, we intend to borrow the successful case-frame ellipsis resolution methods developed recently In XCALIBUR [5] LanguageCraft [10], and PSLI-3 [7], and integrate them Into the universal parser architecture. These methods rely primarily on case-frame semantics and on functional properties of the syntax. Second, we expect to work on extending and applying the embryonic work on practical anaphora resolution in XCALIBUR and work on handlIng metalinguistic utterances [6]. Third, we will focus attention on default inference processes to fill In Implicit Information lacking In the source text, but required for accurate translation. Such information includes subjects in Japanese, which are optional when inferable from context, but which must be stated explicitly in translating to English. At present we utilize a handful of ad-hoc rules to supply default subjects, levels of politeness, etc., but a more principled and systematic approach is required. Fortunately, the universal parser architecture provides an ideal computational framework Into which new knowledge sources may be introduced. And, the knowledge-based translation task provides oopious and severe empirical tests for our theoretically-insepired ideas and methods.

## 5. References


1. Bresnan, J. and Kaplan, R., *Lexical-Functional Grammar: A Formal System for Grammatical Representation,* MIT Press, Cambridge, Massachusetts, 1982, pp. 173-281.
2. Carbonell, J. G., Cullingford, R. E. and Gershman A. G., "Steps Towards Knowledge-Based Machine Translation," *IEEE Trans. PAMI,* Vol. PAMI-3, No. 4, July 1981.
3. Carbonell, J. G. and Joseph, R., "The FrameKit Reference Manual", CMU Computer Science Department internal paper.
4. Carbonell, J. G., and Tomita, M "Knowledge-Based Machine Translation, The CMU Approach," in *Machine Translation: Theoretical and Methodological Issues,* Nirenberg, S., ed., Cambridge, U. Press, 1986.
5. Carbonell, J. G., Boggs, W. M., Mauldin. M. L. and Anick, P. G., "The XCALIBUR Project, A Natural Language Interface to Expert Systems and Data Bases," in *Applications in Artificial Intelligence,* S. Andriole, ed., Petrocelli Books Inc., 1985.
6. Carbonell, J. G., "Meta-Language Utterances In Purposive Discourse," Tech. report, Carnegie-Mellon University, Computer Science Department, 1982.
7. Frederking, R. E., *Natural Langauge Dialog in an Integrated Computational Model,* PhD dissertation, Carnegie-Mellon University, Computer Science Department, 1987.
8. Hayes, P. J., Hauptmann, A. G., Carbonell, J. G. and Tomita, M., "Parsing Spoken Language: A Semantic Caseframe Approach," *11th International Conference on Computational Linguistics (COLING86),* Bonn, U.K., August 1986.
9. Kay, M., "Functional Unification Grammar: A Formalism for Machine Translation," *10th International Conference on Computational Linguistics,* Stanford, July 1984, pp. 75-78.
10. Carnegie Group Inc., *The LanguageCraft Reference Manual,* Pittsburgh, PA, 1985.
11. Lytinen, S., *The Organization of Knowledge in a Multi-lingual, Integrated Parser,* PhD dissertation, Yale University, 1984.
12. Nirenburg, S., Raskin, V and Tucker, A., "On Knowledge-based Machine Translation," *Proceedings of the International Conference on Computational Linguistics,* COLING86, Bonn, West Germany, August 1986, pp. 39-51.
13. Shieber, S. M., "Using Restriction to Extend Parsing Algorithms for Complex-Feature-Based Formalisms," *23rd Annual Meeting of the Association for Computational Linguistics,* Chicago, July 1985, pp. 145-152.
14. Tomita, M., *Efficient Parsing for Natural Language: A Fast algorithm for Practical Systems,* Kluwer Academic Publishers, Boston, MA, 1985.
15. Tomita, M., "An Efficient Context-free Parsing Algorithm for Natural Lnaguaaes," *9th International Joint Conference on Artificial Intelligence (IJCAI85),* August 1985.
16. Tomita, M. and Carbonell, J G., "Another Stride Towards Knowledge Based Machine Translation," *11th International Conference on Computational Linguistics (COLING66),* Bonn, August 1986.
17. Tomita, M. and Carbonell, J. G., "The Universal Parser Architecture for Knowledge-Based Machine Translation," Technical Report, Center for Machine Translation, Carnegie-Mellon University, 1987.


## I. Sample runs of the Universal Parser and Generator

All of the following example sentences were parsed and generated (translated) in 1 to 6 seconds using the first-generation compiled run-time grammars, which ignore capitalization and do not require punctuation. Each example starts with the source text, shows the internal semantic representation, and concludes with the corresponding target text. Japanese is typed in Romaji and converted automatically and unambiguously into Kanji as it is parsed.

### 1.1. Englsh-to-Japanese Translations

```
>take the medicine with thraa glasses of water
 •vary six hours for two days

(((rCTNAME *INGEST-MEDICINE) (:MOOD IMP)
  (:DURATION ((:CTNAME *DURATION) (:DAY 2)))
  <:TREQ
   ((•.CFNAME *FREQUENCY)
    (:INTERVAL ((rCTNAME *DURATION) (:HOUR 6)))
    (:TIMES 1)))
  (: XNGE8T-WTTH
   ((rCTNAME *DRXNKXNG-WATER)
    (:QUANT ((rCTNAME *TOOD-QUANTITY)
             (:GLASS 3)))))
  (:OBJECT ((:CFNAME *MEDICINE))))
```

```
>whan i walk i hava a pain in tha left foot

(((rCTNAME 'HAVE-A-SYMPTOM) (:MOOD DEC)
  (.ASSOCIATED-ACTION
   ((rCTNAME *WALK) (:MOOD DEC)
    (:AGENT
     ((:CFNAME *PATIENT)
      (:HUMAN +) (:PRO +) (:NUMBER SG) (.PERSON 1)))
    (:TIME PRESENT)))
  (: AGENT
   ((rCTNAME *PATIENT)
    (:HUMAN +) (:PRO+) (:NUMBER SG) (rPERSONI)))
  (:TIME PRESENT)
  (:SYMPTOM
   ((rCTNAME *PAIN)
    (:LOCATION
     ((rCTNAME *BODY-PART)
      (:CONTRASTIVE-SPEC((:CFNAME 'DIRECTIONAL-SPEC)
                         (:DIRECTION 'LEFT)))
      (:NAME *FOOT)))))))
```

### 1.2. Japanese-to-English Translations

```
>kyounannokusurionominasitaka


(((:CFNAME *XNGEST-MEDXCINE) (:MOOD QUES)
  (:WHEN
   ((rCTNAME *TIME) (:WH -)
    (:TIME-RELATION
     ((:CFNAME *TIME-RELATION)
      (:INTERVAL ((:CFNAME *DURATION) (:DAY 0)))
      (rDIRECTION -)))))
  (rOBJECT ((:CFNAME *MEDICINE) (:WH +)))
  (:MH +) (:TIME PAST)))

"WHAT MEDICINE DID YOU TAKE TODAY**

>sakuyanagahirihirisiina8ita


(((:CFNAME 'HAVE-A-SYMPTOM) (:MOOD DEC)
  (:WHEN
   ((:CFNAME *TIME) (:WH -)
    (:DAY-SEGMENT ((:CFNAME *NIGHT)))
    (:TIME-RELATION
     ((:CFNAME *TIME-RELATION)
      (:INTERVAL ((rCTNAME *DURATION) (:DAY 1)))
      (:DIRECTION -)))))
  (•.SYMPTOM
   ((rCTNAME *PAIN)
    (:LOCATION ((:CFNAME *BODY-PART) (rWH -)
                (:NAME *EYE)))
    (:PAIN-SPEC ((rCTNAME *BURNING)))))
  (-.TIME PAST)))

"I HAD A BURNING PAIN IN THE EYE LAST NIGHT**
```