Controlled Language and Knowledge-Based Machine Translation: Principles and Practice [1]

Eric H. Nyberg, 3rd
Teruko Mitamura
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213
{ehn,teruko}@cs.cmu.edu

Abstract

For applications with certain properties (well-defined technical domain, technical vocabulary, simple grammatical style), use of a controlled language can enhance the accuracy of knowledge-based machine translation (Baker, et al., 1994). In the continuing development of the KANT system (Mitamura, et al., 1991; Mitamura and Nyberg, 1995), we have explored different sublanguage techniques which limit the complexity of natural language analysis, thus increasing the likelihood of an accurate translation. In this presentation, we will describe some of the general techniques that have been developed for sublanguage use within the KANT system.

1 Introduction

In recent years there has been increasing interest in the use of controlled source languages in MT (cf. Adriaens and Schreurs, 1992 and the references cited there). In this paper, we focus on the use of controlled input language in the KANT translation system (Mitamura, et al., 1991). Controlled English is used to improve the clarity of expression in the source text, and to improve the quality of the translation output.

KANT is a knowledge-based, interlingual machine translation system, a descendant of the KBMT-89 system (Goodman and Nirenburg, eds., 1991). KANT uses explicit source language lexicons, grammars and domain semantics to produce an *interlingua representation* (IR) for each sentence. Each IR is a semantic frame containing features and semantic roles, which may be filled by other IR frames. If a source sentence has more than one possible analysis, KANT produces more than one IR structure.

The input to the target language generation module is the set of IRs produced for a source sentence. The decoupling of the analysis and generation phases has

[1]Accepted for presentation at the First International Workshop on Controlled Language Applications, to be held at Katholieke Universiteit, Leuven, March 26-27, 1996.

various advantages, especially for multi-lingual translation, which have been discussed elsewhere (e.g., Mitamura, et al., 1991). In this paper, we will focus on how source language analysis can result in ambiguous (disjunctive) interlingua representations, which in turn can cause accuracy problems in the target language output.

In the remainder of this section, we explain the architecture of the KANT analyzer, ambiguity in source language analysis, and the translation inaccuracies which can result. In Section 2, we describe the use of controlled source language in KANT and how controlled language can improve the accuracy of translation. In Section 3, we discuss how the success of these techniques depends on certain characteristics of the translation domain.

1.1 Unification Grammar and the Tomita Parser

The KANT analyzer is based on the Tomita LR Parser/Compiler, which compiles pseudo-unification grammars into run-time LR parsing tables (Tomita, 1986). The run-time parser is non-deterministic; it uses an efficient packed forest representation to maintain multiple parse trees in parallel. For example, assume that the parser is using a grammar containing these simple phrase structure rules:

```
<s>  <- (<np> <vp>)
<np> <- (<n> <pp>)
<np> <- (<n>)
<vp> <- (<vp> <pp>)
<vp> <- (<v> <np>)
```

When presented with a sequence of tokens matching the categories `N V N PP`, the LR Parser will always return both possible analyses, e.g.:

```
[NP [VP [NP PP]]]
[NP [VP NP PP]]
```

In a typical KANT application, lexical structures can also be non-deterministic. If a source term matches more than one lexical entry, the LR Parser will create a disjunction (OR), representing both readings in parallel. For example, a term like *rip* can have both a general meaning (e.g., *rip a sheet of paper*) and a specific technical meaning (e.g., *rip a pine board*). The same term may also have different parts of speech; in the case of *rip*, we may use the term to refer to a physical state (noun reading) as well as an action (verb reading). If this distinction were available for both senses listed above, then the LR parser would build an OR with 4 disjuncts for the term, e.g.[2]:

[2]In this paper, we use an asterisk (`*`) to denote semantic concepts; `*O` indicates an object concept, and `*A` indicates an action concept.

```
(*OR* ((root ''rip'')(cat n)(sem *O-RIP-THIN-SHEET-OR-CLOTH))
      ((root ''rip'')(cat n)(sem *O-RIP-WOOD-BOARD))
      ((root ''rip'')(cat v)(sem *A-RIP-THIN-SHEET-OR-CLOTH))
      ((root ''rip'')(cat n)(sem *A-RIP-WOOD-BOARD))
)
```

In the subsections that follow, we explore how the non-determinism in the LR parser can lead to serious difficulty in practical translation domains.

1.2 Non-Determinism and Ambiguity

In domains which deal with the physical world, interesting ambiguities can arise. Consider an example from a computer installation manual:

```
Push the button on the lower left of the screen.
```

The topic may be hardware setup, implying a 3-dimensional screen, or software setup, implying a 2-dimensional screen. If an application encodes only one meaning for a source term, then inaccuracy will result in translation if the author uses the term to denote additional meanings (especially when different target terms are required for the different meanings). On the other hand, representing all the possible meanings of terms in any given domain leads to increased costs in lexicon development and increased complexity in source language analysis.

Even if we assume that it is cost-effective to build a complete semantic lexicon which encodes all the possible domain meanings of terms, we are still left with the problem of ambiguity. To return to the previous example, assume an encoding of two meanings for *push* (`*A-MOUSE-SELECT-BUTTON`, `*A-PUSH-BUTTON-SWITCH`), two meanings for *button* (`*O-BUTTON-WIDGET`, `*O-BUTTON-SWITCH`) and two meanings for *screen* (`*O-2D-SCREEN`, `*O-3D-SCREEN`). Even if we use semantic subcategorization restrictions, so that *button* is `*O-BUTTON-WIDGET` only when *push* is `*A-MOUSE-SELECT-BUTTON` and *screen* is `*O-2D-SCREEN`, for the sentence in question there are at least two valid interpretations in the domain:

```
(*A-MOUSE-SELECT-BUTTON
    (PATIENT (*O-BUTTON-WIDGET
                 (LOCATION (*O-2D-SCREEN)))))

(*A-PUSH-BUTTON-SWITCH
    (PATIENT (*O-BUTTON-SWITCH
                 (LOCATION (*O-3D-SCREEN)))))
```

How can the translation system resolve this ambiguity?

1.3 Sources of Inaccuracy in Analysis

Unresolved ambiguity causes a reduction in translation accuracy. Even when there is significant semantic knowledge to be leveraged for the domain, sentences often have more than one possible interpretation (Baker, et al., 1994). Since the KANT system produces a single output sentence for each input sentence without consulting a target language expert, it must resort to weak heuristic methods for selecting a particular representation as input for generation[3], with a potential decrease in the accuracy of the translations produced.

In general, these types of ambiguity must be coped with:

- Syntactic Ambiguity (multiple syntactic analyses)
- Lexical Ambiguity (multiple parts of speech, multiple meanings)
- Referential Ambiguity (WH-forms, WH-movement, pronouns, clitics, etc.)

In the next section, we describe the controlled language techniques used to reduce ambiguity and improve translation accuracy in KANT.

2 Use of Controlled Language

The use of controlled language in KANT falls into two broad categories: lexical control and grammatical control. In this section we also touch on some issues related to the control of referential ambiguity, although this has not been a primary focus in our work.

2.1 Lexical Control

A key element in controlling a source language is to restrict the authoring of texts such that only a pre-defined vocabulary is utilized. In order to define a controlled vocabulary for a particular application domain, pre-existing documents are analyzed as an initial source of vocabulary. This initial vocabulary is further refined as the domain meanings of each term are encoded, and emerging lexical classes begin to collect domain-specific closed-class items (Mitamura and Nyberg, 1995).

Each domain will contain a set of ambiguous terms (words for which the same root/POS pair has more than one semantic assignment). It is important to judge the relative frequency of these terms, and the number of senses they carry. If there are

[3] Possible heuristics include: iterating through the representations to find one which produces a maximal output string; preferring representations using context-based patterns; statistical language models; etc.

many highly-ambiguous terms, then building a knowledge base sufficient to produce only the semantically acceptable interpretations can be costly.

Our experience has been that the single most useful way to improve the accuracy of a knowledge-based MT system is to limit lexical ambiguity (Baker, et al., 1994). The types of lexical control used in KANT can be categorized as follows:

- **Part of Speech**
  Whenever possible, the application should limit the allowable parts of speech for each term to the minimum necessary for adequate expression in the domain. As domains become more general, it is less feasible to make this kind of restriction, with a natural increase in processing complexity and a potential decrease in output accuracy.

  It is also useful to consider whether there are entire categories of lexical items that can be excluded. For example, our experience has been that technical documentation which provides descriptions or instructions generally does not require much use of WH-words or pronominal anaphors, so it is sometimes possible to restrict their use.

- **Other Lexical Features**
  In addition to part-of-speech, lexical items may also carry different sets of lexical features which distinguish their grammatical behavior. It is useful to control this variation when it applies to a single term. For example, the valency of verbs should be restricted to just the subcategorizations that are sensible in the domain. Another example is the restriction of complement clause. It is important to limit types of complement clauses to particular subclasses of verbs (e.g., verbs of causality, e.g., *This motion causes [the flaps to extend].*). Allowable verb arguments are explicitly represented in the lexicon.

- **Limit Meanings per Word per Domain**
  It is important to control the number of senses allowed for each term. In specialized technical domains, it is sometimes possible to limit the meaning of a term to a single sense, excluding even general senses (e.g., the term *flap* can be limited to just its technical meaning in aircraft operations manuals).

- **Semantic Domain Model Restrictions**
  Although not strictly a part of input control, the use of a semantic domain model can also help to limit the parsing complexity during source analysis. As illustrated above in Section 1.2, syntactic ambiguity can be limited via restrictions on the possible fillers of semantic roles in the interlingua representation (for further discussion, see (Mitamura, et al., 1991)).

- **Annotation of the Input**
  In recent years, different markup languages such as SGML and HTML have

made it possible for documents to be 'tagged for content' – the markup used can be defined in a way that specifies the semantic content of the tagged material. In many cases the markup provides additional clues concerning the semantics of potentially ambiguous strings (e.g., alphanumeric codes, numbers, etc.).

It is also possible to annotate the input string with additional information during the initial phase of syntactic analysis. For example, during the analysis phase of the SHOGUN text extraction system, information about the probable assignment of lexical features was appended to surface forms in preprocessing step (Jacobs, et al., 1993). In the TIPSTER task domains (joint venture and microelectronics), SHOGUN used pre-processing to identify names, dates, locations, and other special phrases.

Given these two examples, it is clear that annotation of the input can be performed as part of the normal text authoring process, or as part of a preprocessing phase. Although the latter is not normally thought of as part of controlled language from the authors point of view, it is certainly possible to involve the author in resolving the meanings of open class terms like names and locations using interactive disambiguation (Brown, 1991).

The KANT system supports various kinds of input annotation via the source grammar definition. Our experience has been that the use of annotations depends directly on the particular types of lexical ambiguity that must be resolved in a given application, the kinds of contextual knowledge and author input available, and the feasibility of adding additional processing to the translation system.

- **Technical Terminology**

  Whenever possible, a system should parse longer strings of technical words as single, atomic units of meaning rather than compositionally-derived structures. In addition to the increased potential for attachment ambiguity in a compositional treatment, it is also the case that the meaning of many phrases cannot be easily derived from the meanings of their parts. For example, the phrase *oil pan* should be considered as an atomic concept, especially where a compositional treatment might pick up the "cooking utensil" meaning for *pan.* Phrasal verb-particle constructions such as *abide by* are also easier to analyze if taken as a unit.

  The potential disadvantage of this approach is that the number of phrases which might be represented separately in the lexicon can grow very large in a practical domain. In large-scale applications, it is important to consider automated and semi-automated methods for corpus analysis and lexicon construction (Mitamura, et al., 1993).

- **Technical Symbology**
  In most technical domains, there is a need for a wide variety of units of measure (e.g., *mm*, *lb-ft*), abbreviations, and acronyms referring to machinery or machine components. To encourage consistency in authoring and to reduce the representational complexity of the lexicon, it is important to standardize the use of these types of terms.

- **Orthography**
  Whenever possible, the spelling, capitalization, hyphenation and use of separators (e.g., "-", "/") in domain terms should be standardized.

2.2 Grammatical Control

Another important way to achieve reduced parsing complexity and increased translation accuracy is to control the types of syntactic structures that are allowed in the input text. In KANT, we have focussed on the following methods for controlling syntactic ambiguity:

- **Limit Ambiguous Attachment**
  Technical documentation invariably contains long sentences with multiple prepositional phrases (PPs). PPs are the single most ambiguous construction for a parser with limited access to domain knowledge and dialog context, since they can potentially modify many words in a sentence:

Torque the bolt *with the wrench.*	(main verb)
Select the **model** *with turbocharger.*	(object NP)
The indicator is **red** *in color.*	(adjective)

  In KANT, PP attachment must be licensed by a combination of mapping rules which link grammatical functions (e.g., subject, object, PP) to semantic roles (e.g., AGENT, PATIENT, INSTRUMENT) and domain model frames which specify the acceptable fillers for a given semantic role.

- **Limit Coordinated Structures**
  Another source of grammatical complexity is the use of coordination (connectors like *and* and *or*). The scope of the coordination and how it relates to other attachments (especially PP attachment) can be difficult to determine:

  Check [[the amount of dirt] and [debris in the coolant]].
  Check [[the amount of [dirt and debris]] in the coolant].
  Check [[the amount of [dirt and [debris in the coolant]].
  Check [the amount of [dirt and debris]] in the coolant.

One important way to limit coordination is to disallow distributed readings across the coordination, e.g.,

(Engine oil) and (coolant)
$*$Engine (oil and coolant)
(Push) and (pull the rod).
$*$Push e_i and pull (the rod)$_i$.

In the KANT analysis grammar, we have tried to limit coordination to conjunction and disjunction of full constituents only. The use of distributed readings like those shown above is not allowed.

- **Limit Processing Time/Space for Complex Sentences**
  Another way to control the complexity of the input text is via simple resource bounds. For example, it is possible to set some predetermined amount of time that is available for the analysis of each sentence; if that threshold is reached before processing is finished, then the system may signal that the sentence is too complex and must be re-written. It is also possible to place an overall limit on the space (memory) used for the analysis, either by limiting the number of syntactic analyses or by limiting the total amount of memory required to represent them. The implementation challenge is to find the right threshold, so that only sentences which the user feels are unacceptably complicated are ruled out.

2.3 Referential Ambiguity

Another source of potential ambiguity and parsing complexity is in the analysis of referring expressions such as WH-words, pronouns, and relative clauses, all of which may modify or be related to a non-adjacent constituent in the sentence. In syntactic theory, these are often referred to as *long-distance dependencies*.

Because KANT applications to date have focussed primarily on descriptive and instructive texts in technical domains, there has been little or no need to support full use of long-distance dependencies, since for the most part they are not necessary in concisely authored text where there is no need to ask questions. Limited use of pronouns and relative clauses is sufficient for authoring most technical documentation (Mitamura and Nyberg, 1995).

3 Which Techniques are Useful, and When?

- **Lexical Control in Limited Domains**
  Categorical lexical exclusions work only in certain styles of text. As domains become more general, less is gained from lexical control; the source text will contain uses of terms which don't correspond to encoded meanings, resulting in translation errors. For a given domain, one must quantify the feasibility of limiting lexical ambiguity, the cost of representing all the possible meanings of terms, and the possible accuracy problems that arise if domain senses for terms are too limited or too general.

- **Feasibility of Semantic Model**
  Encoding a large number of semantic restrictions requires a cost-effective combination of:

  - Automated acquisition from corpora
  - Manual encoding
  - Generalization via semantic hierarchies, generalization rules (Mitamura, et al., 1993)

  If there is little existing text to work with and the domain is large, the amount of manual encoding might be prohibitive. On the other hand, if a large amount of text is available and accurate methods are used to extract domain relations, then significant parts of a domain model can be bootstrapped from corpus analysis.

- **Grammar Control**
  Grammatical limitations are feasible only for certain styles of text, e.g., technical information (descriptions and instructions for equipment). In more general domains, where style varies greatly from document to document (e.g., newspaper stories), it is less feasible to make strong limitations in the syntax. The most suitable domains for grammatical control are those where translation is performed as part of disseminating internally-produced documentation. Grammatical control is less feasible when translation is done as part of assimilating information produced at multiple external sites.

- **Complexity Threshold**
  Resource bounds on source analysis can be tricky to implement, due to:

  - Accuracy of the complexity measure. The resource bound should be reached only for sentences that are too complex to be analyzed and translated accurately, and not for frequently-occurring constructions required in the domain.

- Horizon effects. Depending on how the resource bound is set, some sentences that should not pass grammar checking may come very close to the threshold and not cross it, resulting in a lack of restriction.

We have found overall that constraining the lexicon seems to achieve the largest reduction in the average number of parses per sentence (Baker, et al., 1994). Limiting the major sources of syntactic ambiguity (PP attachment, coordination) is also important, unless these constructions are scarce in the domain. For all of the other possible techniques, relative importance increases with the prevalence of the given construction or phenomenon in the domain.

References

[1] Adriaens, G. and D. Schreurs (1992). "From COGRAM to ALCOGRAM: Toward a Controlled English Grammar Checker," *Proceedings of COLING-92.*

[2] Baker, K., A. Franz, P. Jordan, T. Mitamura and E. Nyberg (1994). "Coping With Ambiguity in a Large-Scale Machine Translation System", *Proceedings of COLING-94.*

[3] Brown, R. (1991) "Automatic and Interactive Augmentation", In K. Goodman and S. Nirenburg (eds), *The KBMT Project: A Case Study in Knowledge-Based Machine Translation*, San Mateo, CA: Morgan Kaufmann.

[4] Goodman and Nirenburg (eds.) (1991). *A Case Study in Knowledge-Based Machine Translation*, San Mateo, CA: Morgan Kaufmann.

[5] Jacobs, P., G. Krupka, L. Rau, M. Mauldin, T. Mitamura, T. Kitani, I. Sider, and L. Childs (1993). "GE-CMU: Description of the SHOGUN system used for MUC-5", *Proceedings of the Fifth Message Understanding Conference.*

[6] Mitamura and Nyberg (1995). "Controlled English for Knowledge-Based MT: Experience with the KANT System", *Proceedings of TMI-95.*

[7] Mitamura, Nyberg and Carbonell (1991). "An Efficient Interlingua Translation System for Multi-lingual Document Production", *Proceedings of the Third Machine Translation Summit.*

[8] Mitamura, Nyberg and Carbonell (1993). "Automated Corpus Analysis and the Acquisition of Large, Multi-Lingual Knowledge Bases for MT", *Proceedings of TMI-93.*

[9] Tomita, Masaru (1986). *Efficient Parsing for Natural Language*, Boston, MA: Kluwer.