Source Language Diagnostics for MT

Teruko Mitamura, Kathryn Baker, David Svoboda, and Eric Nyberg
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{teruko,klb,svoboda,ehn}@cs.cmu.edu

Abstract

This paper presents a source language diagnostic system for controlled translation. Diagnostics were designed and implemented to address the most difficult rewrites for authors, based on an empirical analysis of log files containing over 180,000 sentences. The design and implementation of the diagnostic system are presented, along with experimental results from an empirical evaluation of the completed system. We found that the diagnostic system can correctly identify the problem in 90.2% of the cases. In addition, depending on the type of grammar problem, the diagnostic system may offer a rewritten sentence. We found that 89.4% of the rewritten sentences were correctly rewritten. The results suggest that these methods could be used as the basis for an automatic rewriting system in the future.

1 Introduction

In recent years, researchers in academia and industry have explored the use of Controlled Language (CL) to improve the input to machine translation. CL is intended to promote clearer writing in a variety of contexts, primarily in the creation of technical text (Huijsen, 1998; Knops & Depoortere, 1998; Means & Godden, 1996; Moore, 2000; Wojcik et al., 1998). Improving a text through the use of CL will also improve the quality of any translations of that text, whether the translation is to be done by humans or machines (Nyberg, Mitamura and Huijsen, 2003). A recent study on evaluation of English to Spanish translation (Torrejon and Rico, 2002) shows that a controlled text obtained a better translation score (0.45) than an uncontrolled text (0.72) using the J2450 Translation Quality Metric from the Society of Automotive Engineering (SAE, 2001).

Although controlled language texts are easier to understand and help to promote higher accuracy in translation, it can be difficult for an author to determine how to rewrite an existing sentence to conform to the rules of controlled language. A controlled language checker which provides automatic feedback to the author is an important tool for efficient authoring (Kamprath, et al., 1998). If a sentence does not conform, then the controlled language software should provide a detailed diagnostic message, and possibly an alternate phrasing which conforms to the CL. In this paper we explore the use of unification parsing with pattern matching to provide diagnostic feedback to the user.

The use of parsing and/or pattern-matching for grammar diagnosis is not new. Previous research efforts that applied parsing and/or pattern matching for grammar and style checking include (Ravin, 1993; Adriaens, 1994; Schmidt-Wigger, 1998; Holmback, et al., 2000). The goal in grammar diagnosis is to identify problematic sentences and provide some feedback to the user on how to correct them.

The KANT system (Knowledge-based, Accurate Natural-language Translation) (Mitamura, et al., 1991; Nyberg and Mitamura, 1996), combines the use of a controlled language for source documents with a unification-based parser that checks to see if the input sentences conform to the controlled language. The original version of the KANT Controlled Language Checker provided limited feedback, in the form of messages flagging unknown words, lexical ambiguities, structural ambiguities, and sentences which could not be parsed by the system. In cases where an input sentence did not conform to the

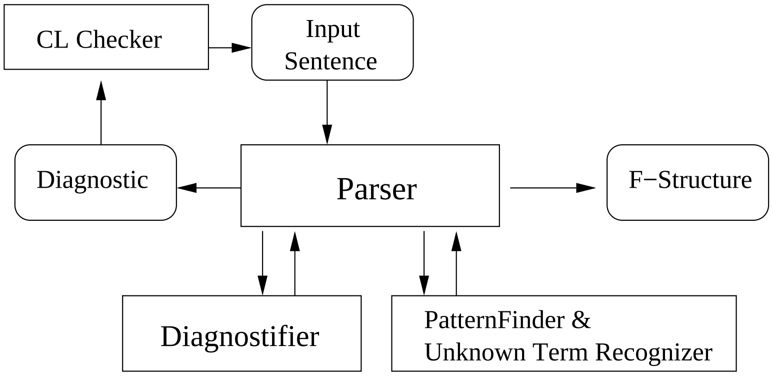

Figure 1: **Diagnostic System in the KANTOO Architecture**

KANT controlled language grammar, the system did not provide any additional diagnostic information regarding the cause of the problem, but simply asked the author to rewrite the sentence. This led to difficulties for inexperienced authors, who could not grasp why a sentence failed to parse and tried several different rewrites in an attempt to get that sentence to pass.

To further improve author productivity, a new set of diagnostics has been added. These diagnostics recognize certain problems with the input sentence and provide detailed diagnostic messages for the author. There are two basic types of diagnostic messages: a) those that offer only an indication of the problem, with the assumption that the author will make manual corrections; and b) those that offer both an indication of the problem and a rewrite that "fixes" the problem, with the assumption that the user will either select the offered rewrite or manually edit the sentence. For example, if a complementizer is missing in a sentence, the system will add the complementizer in a proposed rewrite.

In this paper, based on an analysis of empirical data drawn from authoring log files, we identify the areas where detailed diagnostics would be helpful. We then describe the design and implementation of the diagnostic system and discuss the results of testing the diagnostic system. We conclude with a discussion of ongoing and future work in this area.

2 Grammar Diagnostics

In order to determine which grammatical issues to diagnose, we studied a set of logs derived from authoring sessions in the domain of heavy machinery. We assessed the frequency with which the authors tried to use various constructions which are outside the CL. Based on frequency, we targeted those constructions which, if diagnosed, would have the greatest positive impact on author productivity.

The log files contained 180,402 entries. Each entry corresponded to a single checking event, in which the author was trying to resolve issues with a single sentence in order to have it pass the controlled language checker. The vast majority of these sentences (94%) passed the checker on the first attempt and did not require rewriting. However, 1461 sentences (0.8%) required 4 or more rewrites before the sentence would pass the checker. Since the sentences falling in this range were the most likely to cause frustration and loss of author productivity, we decided to address the worst 0.8% in this study - a set of 1461 sentences from the original log files. We first examined the log files by hand, trying to determine the source of the problem when large numbers of rewrites were attempted by the author.

We also analyzed a set of documents from a different domain (laser printer user manuals) to see if the same types of problems would exist. From the two different types of domains, we found that the following problems were most common, and that diagnosing these problems with specific feedback to the author would probably be the most beneficial for author productivity:

- Unknown Noun Phrase: Although the KANT CL Checker checks for unknown single words before parsing each sentence, it does not check for unknown nominal compounds. Since the KANT CL does not allow arbitrary noun-noun compounding, more specific feedback to the

Diagnostics with no Default	**Reason**
COORDINATED_ADJ	rewrite depends on the conjunction. *smooth and shiny* vs. *smooth or shiny*. future work.
ELIDED_NP	need to determine which NP to insert.
IMPROPER_ING	the correct re-write might be reduced relative clause (*the X that is V-ing*), subord. clause (*while X is V-ing*), etc.
LIKE	might be able to use *as*. future work.
UNKNOWN_NP	lexicographer needs to review the terms.
PARENS	need to move the parenthetical element. future work.
WHEN_VING	need to refer back to the subj. of the main clause. future work.

Figure 2: **Diagnostics with No Default**

Diagnostics with Default	**Format**
IMPROPER_PUNC	punctuation is removed and/or replaced
MISSING_DET	determiner *the* is inserted
IN_ORDER_TO	*in order* is inserted before *to*
MISSING_PUNC	appropriate punctuation is added
MISSING_THAT	word *that* is added
MISSING_COMMA	comma is added
BY_USING	word *by* is inserted before *using*
IF_WHETHER	*if* is replaced by *whether*

Figure 3: **Diagnostics with Default**

author would be helpful. We found that the author often tries to rewrite the whole sentence without realizing that the problem is just an invalid nominal compound.

- Missing Determiner: The use of determiners in noun phrases is strongly recommended in KANT Controlled English (KCE). We found that authors often omit determiners inside sentences.

- Coordination of Verb Phrases: Coordination of single verbs or verb phrases is not allowed in KCE, since the arguments and modifiers of conjoined verbs may be ambiguous for translation.

- Missing Punctuation or Improper Use of Puctuation: The author may omit required punctuation, or make inconsistent use of punctuation marks such as comma, colon, semicolon and quotation.

- Missing "in order to" phrase: If an infinitival verb phrase is used to indicate purpose, KCE strongly recommends that the author writes "in order to" instead of "to". For example, "Click on the button to receive the channel settings" should be rewritten: "Click on the button in order to receive the channel settings".

- Use of "-ing": In KCE, the "-ing" form cannot be used immediately after a noun. For example, "The engine sends the information indicating that the engine RPM is zero" must be rewritten as: "The engine sends the information that indicates that the engine RPM is zero".

- Coordination of Adjective Phrases: In KCE, adjective coordination before a noun is not allowed because it may introduce ambiguity. For example, "top left and right sides" must be rewritten as "the top left side and the top right side".

- Missing Complementizer, "that": The complementizer "that" cannot be omitted in KCE. For example, "Ensure it is set properly" must be rewritten as "Ensure that it is set properly".

We implemented grammar diagnostics for each of these high-priority problems. To the above list of most frequent problems, we added other useful diagnostics for problems such as use of contraction (e.g. "where's"). The design and implementation of the diagnostic system are described in the next section.

3 Design and Implementation

The structure of the KANTOO diagnostic system is shown in Figure 1. The Parser operates on each input sentence, trying to create an F-Structure which represents the parse tree of the sentence. Our grammar has the ability to recognize common errors (such as omitting ”the” before a noun). When the grammar recognizes a common error, it builds the F-Structure as if the error had not occurred, and inserts a diagnostic message describing the error into the F-Structure. The result is an F-Structure that may contain one or more diagnostics.

The result of the Parser is passed to the Diagnostifier module. The Diagnostifier's job is to find any diagnostics the grammar may have inserted into the F-Structure, and determine which diagnostic (if any) should be displayed to the author. For example, a sentence containing an ambiguous term might have 2 F-Structures, one with a verb reading for the term and one with a noun reading plus a missing determiner diagnostic. In this case, the Diagnostifier will prefer the F-Structure containing the verb reading (and no diagnostics), and return only that F-Structure. If the Diagnostifier returns an F-Structure without any diagnostics, the Parser returns an “OK” to the CL Checker. Otherwise, the Parser returns the diagnostic indicated by the Diagnostifier.

Occasionally our grammar will fail to parse a sentence because it contains errors that the grammar cannot recognize. (A simple example of such an error is a sentence that contains unknown terms.) The Parser sends such sentences to the Patternfinder module. The Patternfinder checks the sentence for various problematic patterns, and if one is found, the Patternfinder returns an appropriate diagnostic to the Parser. In addition to searching for unknown terms, the Patternfinder will search for patterns which are known to be invalid. Some example patterns include ellipses (”...”) and contractions (”aren’t”, ”can’t”, etc). If the Patternfinder cannot find a pattern match for a failed sentence, the Parser returns a general error that indicates to the author that the sentence is not grammatical.

3.1 Pattern Matching and the Parsing Architecture

One characteristic of the diagnostic rules located in the grammar is that the sentence must parse completely in order for these rules to apply. However, there are certain constructions, such as contractions, which are outside the controlled language, regardless of the sentence. The Patternfinder will match a sentence against a set of raw patterns and send a message to the author in case one of the patterns matches. This provides additional rewriting help with little overhead, and with no disruption to the parsing grammar. This also minimizes the level of complexity in the grammar. An example pattern is the following pattern for a semicolon:

```
[";"] =
((type SEMICOLON)
 (message "Do not use ’;’.
  Semicolon is not part of KCE."))
```

The pattern that matches is a semicolon character, and a message to the user is provided. In some cases, a suggestion for rewriting can also be offered. We detail this in the following section.

Since the grammar diagnostics are incorporated into a full parse of a sentence, which is the desired output form, pattern matching follows the parser. If no parse is available for a sentence, then we see if the sentence might match one of the patterns that are problematic for the grammar.

3.2 Types of Diagnostics

The purpose of diagnostic rules or patterns is to provide information to the author. The diagnostics can be divided into two categories. The first type of diagnostic gives a message which tells why a sentence is not part of the CL. The second type of diagnostic provides a similar message, but also offers a default rewrite for the sentence. The author can select the rewrite or can choose to ignore it and rewrite the sentence in another way. Below we discuss the rationale for each type of diagnostic, and provide examples.

The first type of diagnostic is a diagnostic message. One example of this type of diagnostic is the UNKNOWN_NP diagnostic. For this diagnostic, the system informs the author that a particular noun phrase is not in the dictionary. The author may want to tag the term as a candidate for the terminology addition process. By this process, a lexicographer decides whether to add the term to the lexicon. Alternatively, the term may be inappropriate for the lexicon. Since no determination of this can be made automatically, it is left to the author to determine what to do with vocabulary items that are not recognized by the parser.

Another diagnostic which does not have a default

Pattern Matching with no Default	**Reason**
QUOTES	too many different uses of quotes.
SEMICOLON	don’t know whether should be comma or period. Some cases are in the grammar as IMPROPER_PUNC. Pattern matcher picks up the other cases.
REFLEXIVE	can’t identify a default.
ELLIPSIS (. . .)	can’t identify a default. Some cases are in the grammar. Pattern matcher picks up the other cases.
DASH	not enough data to support
LOOK_LIKE	not enough data to support

Figure 4: **Pattern Matching without Defaults**

Pattern Matching with Default	**Format**
CONTRACTION	expand the contraction, e.g. *haven’t* to *have not*, *you’re you are*, etc.
WHETHER_OR_NOT	change to *whether*
HAVE_TO	change to *must*
ONE_ANOTHER	change to *each other*

Figure 5: **Pattern Matching with Defaults**

rewrite is the IMPROPER_ING diagnostic. This diagnostic fires when an *-ing* form appears directly after a noun. There is more than one way to rewrite this form. The participle could be the verb in a relative clause, as in *customers using printers in dusty environments* (means customers who are using printers), or it could be a subordinate clause, e.g. *print the user guide using your printer* (means to print the user guide by using your printer). Currently, the diagnostics are handled as syntactic constructions without additional semantic knowledge. Unless the rewritten form is very clear to the parser, we do not want to assign a default. Future work might include accessing the KANT domain model, which contains semantic roles. For example, one might be able to restrict the subject candidates for a verb, in the case of the *-ing* diagnostic mentioned above. Figure 2 contains a list of the diagnostics for which we do not assign a default rewrite.

For many diagnostics, we are able to suggest a rewrite. This occurs in the cases where the diagnostic is narrowly defined. The author’s error is easily correctable by the addition or removal of a particular word or punctuation mark (see Figure 3).

In the case of pattern matching, the system provides a message indicating the problematic part of a sentence, and optionally can suggest a rewritten form. We use the same criteria for deciding whether a pattern should have a default. For example, one pattern that does not have a default associated with it is the quotation marks pattern. Quotes which reference another part of a document, e.g. *Go to ”Printer Software” on page 50*, may be rewritten with a specific tag. Some quotes may simply be removed, as in the case of scare quotes, e.g. *a parallel cable with a ”C” connector*. Other quotations must be rewritten in some other way. In contrast, in the case of contractions, we can use the expanded form of a contraction as a good default rewrite. Figures 4 and 5 list the patterns which have no rewrites associated with them, and those with default rewrites.

4 Evaluation

We tested the diagnostic system on a set of original documents from computer printer manuals, which were not written to conform with KCE. We tested a total of 6507 sentences and found that 2278 sentences (35%) conformed to KCE. The low acceptance rate was partly due to the omission of required XML tags in the original texts. When we tagged a subset of the texts, which contained 1347 sentences, 62% of the sentences (837 sentences) passed KCE.

We examined the sentences which did not conform with KCE. We tested a total of 4229 non-KCE sentences and found that 2843 sentences (67.2%) received a diagnostic message from the system. Of the 2843 sentences diagnosed, 1741 sentences (60%) produced one or more of the grammar diagnostic messages listed in a previous section, and 1129 sentences (40%) contained unknown single terms.

Diagnostic	**No. Sentences**	**No. Correct**	**% Correct**
UNKNOWN_TERM	234	234	100%
UNKNOWN_NP	158	140	88.6%
IMPROPER_PUNC	134	122	91%
MISSING_DET	61	22	36.1%
IN_ORDER_TO	43	35	81.4%
MISSING_PUNC	35	33	94.3%
MISSING_THAT	19	18	94.7%
IMPROPER_ING	7	7	100%
MISSING_COMMA	11	11	100%
WHEN_V-ING	5	4	80%
ADJ_COORD	4	4	100%
PARENTHESIS	2	2	100%
BY_USING	7	7	100%
ELIDED_NP	3	2	66.7%
IF_WHETHER	1	1	100%
QUOTES	42	42	100%
CONTRACTION	61	61	100%
SEMICOLON	4	4	100%
REFLEXIVE	3	3	100%
ELLIPSIS	2	2	100%
HAVE_TO	1	1	100%
Total	837	755	90.2%

Figure 6: **Results for Each Diagnostic**

We conducted a further examination on a randomly-selected subset of the documents to measure the correctness of the diagnostics. We tested 1437 non-KCE sentences and found that 837 sentences (58.2%) received some type of diagnostic message from the system. Of the 837 sentences diagnosed, 755 sentences (90.2%) were diagnosed correctly. When we examined just the grammar diagnostics, we found that 521 sentences out of 603 (86.4%) received correct grammar diagnostic messages. Figure 6 contains the results for each diagnostic.

We found that the diagnostic for missing determiners was the most difficult to implement precisely, and the accuracy of this diagnostic was only 36.1% in the evaluation. We further examined the failures, and found that there are some sentences which require XML tags instead of a determiner on a noun phrase (e.g., for a menu item in the document). In other cases, we found idiomatic expressions which do not require a determiner (e.g. “from side to side”). Also, titles that are noun phrases do not require a determiner.

We also examined the diagnostics which offer a rewrite. There were 312 sentences out of 603 which fell into this category. We identified the diagnostic messages containing a default choice which were correct. Of the 289 sentences correctly diagnosed, 279 sentences (96.5%) offered a correct rewrite. If we measure all the diagnostics which offer default rewrites (312 sentences), then accuracy is measured at 89.4%. This result implies that an automatic rewriting system that fixes problems without asking the author might achieve around 90% accuracy.

5 Discussion and Future Work

In this paper, we described the empirical analysis of a large set of sentences from laser printer user manuals. We described a new diagnostic system that recognizes problems in the text and provides specific diagnostic messages to the author. In an experiment with non-KCE sentences, the diagnostics correctly identified the problem for 90.2% of the sentences. The accuracy of automatic rewrites was 89.4%, for sentences where the system offered a rewrite.

In the future, we would like to develop a process which will further improve author productivity by incorporating automatic rewriting into the CL checker. As mentioned in the previous section, some diagnostics and rewrites are more accurate than others. For example, the missing comma rewrite seems to be very accurate, while the missing determiner diagnostic is quite inaccurate. The implication is that some diagnostics require further improvement before rewrites can be applied automatically.

Another important topic for ongoing research is author acceptance of automatic rewriting. It is not clear to what degree the author is willing to grant autonomy to an automatic rewriting system. Perhaps there are some rewrites which can always be automatic; others that may be selectively enabled by certain authors; and yet others which will always be interactive due to the general difficulty of correct diagnosis. Future work should address the tradeoffs between system autonomy, productivity, and some measure of document quality.

Bibliography

Adriaens, G. (1994). “Simplified English Grammar and Style Correction in an MT Framework: The LRE SECC Project”. In *Proceedings of the 16th Conference on Translating and the Computer.*

Holmback, H., L. Duncan and P. Harrison (2000). “A Word Sense Checking Application for Simplified English”. *Proceedings of the Third International Workshop on Controlled Language Applications (CLAW 2000)*, Seattle, Washington.

Huijsen, W. O. (1998). “Controlled Language - An Introduction”. *Proceedings of CLAW 1998*, Pittsburgh.

Kamprath, C., T. Mitamura and E. Nyberg (1998). “Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English,” *Proceedings of the Second International Workshop on Controlled Language Applications*, Pittsburgh, PA.

Knops, U. and B. Depoortere, (1998). “Controlled Language and Machine Translation”. *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW-98)*, Pittsburgh, PA.

Means, L. and K. Godden (1996). “The Controlled Automotive Service Language (CASL) Project”, *Proceedings of the First International Workshop on Controlled Language Applications (CLAW-96)*, Leuven, Belgium.

Mitamura, T. (1999). “Controlled Language for Multilingual Machine Translation”. *Proceedings of Machine Translation Summit VII*, Singapore.

Mitamura, T., Nyberg, E. and Carbonell, J. (1991). “An Efficient Interlingua Translation System for Multi-lingual Document Production”. *Proceedings of Machine Translation Summit III*, Washington, DC.

Moore, C. (2000). “Controlled Language at Diebold, Incorporated”. *Proceedings of the Third International Workshop on Controlled Language Applications (CLAW-2000)*, Seattle.

Nyberg, E., T. Mitamura and W. Huijsen (2003). “Controlled Language,” in H. Somers, ed., *Computers and Translation: Handbook for Translators*, Johns Benjamins.

Nyberg, E. and T. Mitamura (1996). “Controlled Language and Knowledge-Based Machine Translation: Principles and Practice”. *Proceedings of the First International Workshop on Controlled Language Applications (CLAW-96)*, Leuven, Belgium.

Ravin, Y. (1993). “Grammar Errors and Style Weaknesses in a Text-Critiquing System” in K. Jensen, G. Heidorn and S. Richardson (eds.) *Natural Language Processing: The PLNLP Approach*, Kluwer Academic Publishers.

Schmidt-Wigger, A. (1998). “Grammar and Style Checking for German”. *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW-98)*, Pittsburgh, PA.

Society of Automotive Engineering (SAE J2450) in http://www.lisa.org/useful/2001/J2450Practice.pdf

Torrejon, E. and C. Rico (2002). “Controlled Translation: A New Teaching Scenario Tailor-made for the Translation Industry”. *Proceedings of the 6th EMAT Workshop: Teaching Machine Translation*, Manchester, England.

Wojcik, R., H. Holmback and J. Hoard (1998). “Boeing Technical English: An Extension of AECMA SE beyond the Aircraft Maintenance Domain”. *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW-98)*, 114-123, Pittsburgh, PA.