Diagnostics for Interactive Controlled Language Checking

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

Abstract

This paper presents a grammar diagnostic system for controlled language checking. The implemented diagnostics were designed 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.

1 Introduction

Since natural language texts often contain sentences which are both structurally complex and potentially ambiguous, the use of Controlled Language (CL) has been developed to encourage clearer writing in a variety of contexts, primarily in the creation of technical text for commercial domains (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).

The KANT system (Knowledge-based, Accurate Natural-language Translation) (Mitamura, et al., 1991; Nyberg and Mitamura, 1996) uses a controlled language to translate technical documents, such as heavy equipment documentation, computer manuals, automotive manuals and medical record texts (Mitamura, 1999). 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.

The original version of the KANT system 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 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.

In this paper, we describe a grammar diagnostic system that has been incorporated into the KANTOO Controlled Language Checker and provides detailed diagnostic messages for the author. In Section 2, we present an analysis of the empirical data, drawn from authoring log files, which was used to identify the areas where detailed diagnostics would be helpful. In Section 3, we describe the design and implementation of the diagnostic system itself. In Section 4 we discuss the results of testing the diagnostic system, and we conclude in Section 5 with a discussion of ongoing and future work in this area.

2 Empirical Analysis

In general, there are two approaches to controlled language checking. In a proscriptive system, a finite set of negative patterns are matched against the text. If one of the patterns is matched, the offending sentence is flagged for rewriting. In a prescriptive approach, the system incorporates a complete grammar which describes all possible grammatical sentences in the language. The grammar is used in conjunction with a parser to test each input sentence. If a sentence does not parse, it is flagged for rewriting. The KANTOO system is a prescriptive system that utilizes a detailed set of grammar rules (written in LFG) and a variant of the Tomita parser (Tomita, 1986) to perform sentence checking. Since a prescriptive approach does not utilize negative patterns to match specific problems, when a parse fails, the cause of the problem is not immediately apparent. To provide more specific diagnostic information for the author, it would be necessary to extend the existing grammar, so that it would be possible to identify specific problematic structures.

In order to determine which grammatical issues to diagnose, we studied author logs derived from sessions with the authoring tool. 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 impact on author productivity. For example, using a noun phrase that was not in the lexicon was determined to be the authors' most frequent grammatical issue. Although the existing KANTOO system indicates when individual words are not in the lexicon, the issue of ungrammatical phrases had not been addressed.

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%) were resolved with a single re-write by the author. However, 1461 sentences (0.8%) required 5 or more tries before the sentence would pass the checker. The results of the error analysis are shown in Figure 1. The number of tries for 5 or more ranges between 5 and 46. Since the sentence 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.

Rewrites Required	**Total Sentences**	**Percentage**
1	169505	94%
2	5404	3%
3	2792	1.5%
4	1240	0.7%
5+	1461	0.8%
Total	180402	100%

Figure 1: **Number of Rewrites per Sentence**

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 found that the following problems were most common, and that diagnosing these problems with specific feedback to the author would be probably be the most useful extension to the grammar checker.

- Missing 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 author would be helpful. We found that on many occasions, the author tries to rewrite the structure of the 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 the author often omits 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, "Grind all surfaces smooth to create the original rail geometry" should be rewritten: "Grind all surfaces smooth in order to create the original rail geometry".

- 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 by "Ensure that it is set properly".

We implemented grammar diagnostics for each of these high-priority problems. The design and implementation of the required extensions to the KANTOO CL Checker are described in the next section.

3 Design and Implementation

In the original KANT CL Checker, a very general message was provided when a sentence failed to pass the checker ("There is a grammatical issue with this sentence."). In order to implement more specific diagnostic messages for the user, a new module was added to the KANTOO architecture: the Diagnostifier (see Figure 2).

The Diagnostifier determines whether or not a particular sentence triggered certain diagnostic patterns in the controlled grammar. If this is the case, then a detailed message is prepared which is then transmitted to the Controlled Language Checker. Depending on the type of diagnostic message, a specific dialog is invoked with the user to address the issue at hand. In the remainder of this section, we present additional detail regarding the use of diagnostics at runtime, along with a description of how diagnostics are implemented in the controlled grammar.

3.1 Diagnostics at Run Time

The parsing process creates feature structures, or f-structures, which describe the syntactic features of a sentence. We use the Lexical-Functional grammar formalism (Bresnan ed. 1982). If the set of possible parses for an input contains at least one f-structure without diagnostics, then any f-structures with diagnostics are pruned, and the parse continues to the Disambiguation module. If all f-structures are diagnostic f-structures, then they are passed to the Diagnostifier. This module is an extension to the parser. The scores of all diagnostics within each f-structure are summed. The f-structure with the lowest total score is preferred. In case of a tie, the system arbitrarily chooses an f-structure.

If the best f-structure has more than one diagnostic, then the diagnostic with the lowest score is presented to the user first. Else, one is chosen arbitrarily. The Diagnostifier creates from the preferred f-structure an ordered pair contaning the f-structure and a diagnostic message,

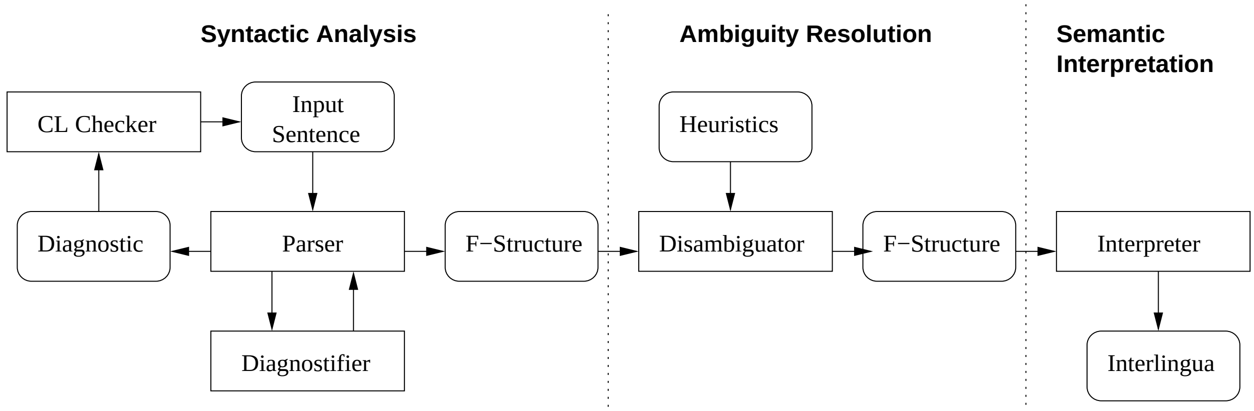

Figure 2: **Integration of Diagnostics in the KANTOO Architecture**

in a format understood by the CL Checker. The message is displayed to the author, and no further processing of the sentence takes place until the sentence is rewritten. In some cases, such as the addition of a period to the end of the sentence, an automatically rewritten version of the sentence is suggested and can be selected in the user interface. Otherwise, the author rewrites the sentence by hand, or marks a term for consideration as an addition to the lexicon.

In figure 3 we show the dialogue window for the sentence `The new refill capacity is 10 gallons with an oil filter change.` The CLC responds that the phrase *oil filter change* is not in the dictionary.

In Figure 4 we show the message that the Analyzer returns for the input sentence `Click on the button to retrieve the channel settings.` The parser has signalled that `in order` can be inserted in front of `to`, to mark the infinitive unambiguously as a purpose clause. In this case the button marked **Fix Sentence** gives the user an opportunity to select the rewritten version that is displayed and have it automatically inserted into the document. **Do Nothing** means no edits are carried out automatically, and the author will rewrite the sentence manually. **Help** brings up background information on the particular diagnostic.

3.2 Diagnostics in the Controlled Grammar

The purpose of grammar diagnostics is to indicate which part of an input sentence does not conform to the CL. We have implemented diagnostics in the controlled language grammar as negative rules. Rules matching an ungrammatical construction create a typical f-structure, but with the addition of a feature local to that structure which contains diagnostic information. The sub-structure stored in this diagnostic feature includes the name of the diagnostic, a message to print to the author, and a score. The score helps to resolve ambiguities. In case a sentence can be rewritten by adding text, insertion text for the rewrite and a pointer to the place of insertion are also included.

For example, we show below a rule which matches any full sentence which is lacking a final period. The diagnostic features in the rule indicate that a period should be inserted at the end of the sentence.

```
(<kce-sentence> <== (<sentence> )
  ((%x0 = %x1)
   (*try*
    ((%(x0 cat) =c sp))
    ((%(x0 diagnostic type) = MISSING_PUNCT)
       (%(x0 diagnostic message) =
         "If this is a sentence, it requires
          a final period.")
       (%(x0 diagnostic insertion-text) = ".")
       (%(x0 diagnostic after) = +)
       (%(x0 diagnostic score) = 13)))))
```

Each diagnostic carries an integer score. The relative scores of the diagnostics were determined

Figure 3: **Diagnostic message when NP is not in the dictionary**

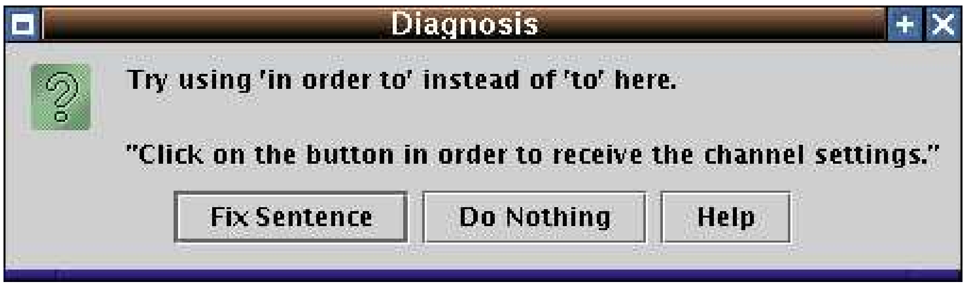

Figure 4: **Diagnostic Message for Interactive Rewriting**

via trial and error, with a tendency to assign more specific diagnostics lower, or better, scores. Sentences with diagnostics are scored such that the most common diagnostic (the "missing NP" diagnostic) is scored high, or near last. The algorithm for finding a noun phrase outside the vocabulary attempts to make any noun-noun compound from consecutive nouns in the dictionary. Since many nouns are ambiguous with other parts of speech, such as adjectives or verbs (e.g. `check`, `view`, `minimum`) we attempt to resolve other possible errors first, reducing the possible ambiguity on this construction.

An exception is that the score of a missing NP is lowered if the NP is standalone. This means that we assume a higher probablity that a standalone phrase is a noun rather than a sentence missing a period. For example, `check valve` is presumed to be a noun, rather than a sentence with a missing determiner on `valve` and a missing period. The diagnostic for diagnosing improper use of NP is scored last, since we must assume first that `-ing` forms are part of a missing NP. Else, they are participial verbs. An example of this is the phrase *temperature setting*, in which we want `setting` to be construed as a noun, not as a present participle. The complete set of diagnostics and their scores is shown in Figure 5.

4 Evaluation

In order to test the implemented diagnostics, we selected 1302 sentences from the author log files which do not conform to KCE for various reasons. The number of sentences is smaller than the number reported in Figure 1, due to recent system improvements that were completed after the initial analysis was performed.

We found that 569 sentences (44%) received a diagnostic message from the system. 415 sentences (32%) exhibited one or more of the issues listed in Figure 5, and 154 sentences (12%) contained unknown terms, which were diagnosed by the existing single-term lexical check and not passed to the Diagnostifier. Of the 569 sentences diagnosed, 35 sentences (6.2%) received an incorrect or misleading diagnostic message; 93.8% of the sentences were diagnosed correctly. Figure 6 contains the results for each diagnostic. We did not find any test sentences that triggered the BY_USING diagnostic.

The remaining 733 sentences in the original

Diagnostic	Description	Score
MISSING_DET	determiner missing before a noun	10 for phrases, else 11
MISSING_NP	noun phrase not in the dictionary	10 if standalone, else 20
IN_ORDER_TO	missing "in order" before "to"	12
MISSING_PUNC	no period at end of sentence	13
BY_USING	need "by" before "using"	15
VP_COORD	two verbs cannot be conjoined	15
MISSING_THAT	use "that" after "make sure"	15
ADJ_COORD	two adjectives cannot be conjoined	16
IMPROPER_PUNC	do not end noun phrase in a period	21
IMPROPER_ING	bad use of an -ing form	25

Figure 5: **Diagnostics and Associated Scores**

Diagnostic	No. Sentences	No. Errors	% Correct
MISSING_NP	240	12	95%
UNKNOWN_TERM	154	0	100%
MISSING_DET	60	14	76.6%
VP_COORD	32	1	96.8%
MISSING_PUNC	27	2	92.5%
IMPROPER_PUNC	25	4	84%
IN_ORDER_TO	15	1	93.3%
IMPROPER_ING	12	1	91.6%
ADJ_COORD	3	0	100%
MISSING_THAT	1	0	100%
Total	569	35	93.8%

Figure 6: **Results for Each Diagnostic**

sample (56%) did not pass the checker but did not raise a specific diagnostic message. On further analysis, we found that a large number of these sentences had problems with SGML tagging (e.g., the author did not change or delete obsolete tags in legacy text that was reused). Other sentences were ungrammatical because they were incomplete (non-sentence) fragments. We also found that comparative sentences were problematic for the authors, and often did not pass the checker. The remaining problems in the test set fell into one of several miscellaneous categories which represented very small fractions of the original set.

5 Discussion and Future Work

In this paper, we described the empirical analysis of a large set of sentences drawn from author log files, which identified a set of writing problems and associated diagnostics that would be most likely to reduce author frustration and improve author productivity. We described a new diagnostic module which detects these problems and provides specific diagnostic messages to the author. In an experiment with non-KCE sentences selected from real author data, we found that the new diagnostic messages correctly identified the specific problem in 93.8% of the cases.

Although the initial accuracy of the diagnostics we implemented is encouraging, there are some remaining issues to be addressed in ongoing and future work:

- *Author Productivity.* We assume that more specific diagnostics and proposed automatic rewrites will improve author productivity, but these effects have not yet been measured in an empirical study. Gains in productivity must also be balanced against potential confusion caused by incorrect diagnostic messages (which may be less useful than a simple indication that the sentence is ungrammatical).

- *Testing Recall.* It is difficult to determine automatically whether there are additional sentences in the test set for which the system should have raised diagnostics, but did not. In this context, less than perfect recall implies that the author will have to rely on the simple "ungrammatical" indication, rather than a specific message. Hence the cost of imperfect recall is mainly one of lost opportunity, rather than incorrectness in the system's behavior. Nevertheless it would be useful to do more work in determining the recall numbers for the various diagnostics, to see if they could be improved.

- *Addressing Other Issues.* In this study we created diagnostics to address only the worst 1% of the sentences in the author log files - those that required 5 or more rewrites to correct. As mentioned in Section 4, there are other issues which could be addressed by more specific diagnostics (e.g., improper SGML tags), but further empirical investigation is required. For instance, it would be interesting to determine the most common problems overall (for all sentences requiring a rewrite), to see whether there is overlap with the existing set of diagnostics, or whether there are other broad categories of issues that could be addressed by additional diagnostics.

Bibliography

Bresnan, J., ed. (1982). *The Mental Representation of Grammatical Models.* The MIT Press, Cambridge, Mass.

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.

Knops, U. and B. Depoortere, (1998). "Controlled Language and Machine Translation". *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW-98)*, pages 42-50, Pittsburgh.

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)*, pages 106-114, Leuven, Belgium.

Mitamura, T. (1999). "Controlled Language for Multilingual Machine Translation". *Proceedings of Machine Translation Summit VII*, pages 46-52, 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*, pages 55-61, Washington, DC.

Moore, C. (2000). "Controlled Language at Diebold, Incorporated". *Proceedings of the Third International Workshop on Controlled Language Applications (CLAW-2000)*, pages 51-61, Seattle.

Nyberg, E., T. Mitamura and W. Huijsen (2003). "Controlled Language," in H. Somers, ed., *Computers and Translation: Handbook for Translators*, to be published by 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)*, pages 74-83, Leuven, Belgium.

Tomita, M. (1986) "Efficient Parsing for Natural Language: a Fast Algorithm for Practical Systems", Kluwer, Boston.

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)*, pages 114-123, Pittsburgh.