Exploiting Knowledge Representation in an Intelligent Tutoring System for English Lexical Errors

Chiu-Chen Hsieh[1], Tzong-Han Tsai[1], David Wible[2], Wen-Lian Hsu[1]
[1] *Institute of Information Science*
Academia Sinica
Nankang, Taipei, Taiwan, R.O.C
Tel: 886-2-27883799x1365
Fax: 886-2-26518660
Email: {gladys, thtsai, hsu}@iis.sinica.edu.tw

[2] *Graduate Institute of Western Languages and Literature*
Tamkang University
Tamsui, Taipei Hsien, Taiwan, R.O.C
Email: dwible@mail.tku.edu.tw

Generally, Intelligent Tutoring Systems (ITSs) rely heavily on the hand-construction and design of the domain knowledge provided for learners. In this paper we attempt to illustrate a central role that knowledge representation can play in automating crucial aspects of ITS design and implementation. We focus on the treatment of English learners' lexical errors. Specifically, we describe a system which exploits highly constrained dialogue with the learners in order to allow a simple algorithm to retrieve the relevant lexical knowledge that the learner needs to remediate the error at hand. For this, we propose a Diagnosis, Interaction and Treatment (DIT) Model for ITS. The entire system relies upon a knowledge representation system (InfoMap) whose structured encoding of English lexical information makes it possible to (1) initiate the relevant dialogue with learners when the system is not sure about learners' intentions, (2) trigger the appropriate lexical knowledge based on their responses, and (3) automatically generate practice and test exercises on this knowledge. Crucially, inferences from the dialogue and automatic practice exercise generation rely on a very general algorithm and the structure of the English knowledge representation, regardless of the ability of the learner. Thus, the system automates what must traditionally be created by hand through human experts.

Keywords: **Intelligent Tutoring System, Error Diagnosis, Knowledge Representation**

1. Introduction

In recent years, Intelligent Tutoring Systems (ITSs) have been commonly used in educating students in various disciplines. In this paper we describe a model called Diagnosis, Interaction and Treatment (DIT), which is distinct from traditional ITSs. We use knowledge representation schemes as in InfoMap (Wu et al., 2002), to handle the entire DIT operation. We illustrate this approach of exploiting knowledge representation by focusing on lexical errors of writers of English as a Second Language (ESL).

Existing approaches to lexical error diagnosis fall into two main categories: spelling checkers and intelligent tutoring systems. The purpose of most spelling checkers is not teaching and learning. They are only for detecting spelling errors and suggesting appropriate spelling. Well-known spelling checkers such as Microsoft Word Spelling Checker and Webster on-line dictionary (http://www-m-w.com) are extremely good at error checking but lack a pedagogical component. Alternatively, intelligent tutoring systems, for the most part, emphasize the student model but only make passing reference to a tutor model. The student models aim to record student responses to pre-set exercises and select teaching materials by hierarchical structures. Error diagnosis in most ITSs is performed by letter-for-letter comparison against an answer key (Heift, 1998; Heift & Nicholson, 2000). According to the ITS literature (Goodkovsky, 1996; Hegarty, 1996), tutor models can be instantiated as an authoring shell (Stankov & Božičević, 1998) that can be accessed through an interface allowing teachers to choose their teaching materials more freely to suit individual student needs.

In traditional approaches, each word and its inflected forms are stored in a database. They match the user input with existing data to detect errors. However, some errors are not easily found in this way. Besides, pedagogical goals cannot be achieved by simply detecting errors. With our approach, which aims to understand student errors, irregular words are stored in a database, as in a traditional spelling checker, but inflection rules are also applied to regular words to get their base forms. In this way, we can deal with the problem found in the following sentence:

How many bussies did you ride?

When normal spelling checkers detect the error “bussies”, they will provide suggestions such as busies, buses and busiest. Whether the learner made a spelling mistake, a typo, or a lexical error is not clear to the system. In contrast, DIT can indicate the possible problems and provide more useful help. Spelling checkers and traditional ITSs do not deal with the conceptual sources of learners’ lexical errors; whereas these are the fundamental parts of DIT’s error diagnosis.

2. The Framework of DIT

The DIT model for lexical errors provides an open writing environment. We allow students to write freely without an exercise model at the beginning. Exercises are helpful for students only when they make mistakes. DIT is responsible for performing the following tasks upon learner input:

- The sentence is parsed into words.
- Each word is checked for lexical errors.
- A dialogue is initiated with the user when ambiguity occurs.
- Appropriate grammatical aids associated with the error type are selected and administered to the student.
- Quizzes are automatically generated by the system from the structured knowledge.
- The above steps can be executed in a cycle, depending on the student’s exercise results.

Figure 1 describes the relationship among all components of the DIT framework.

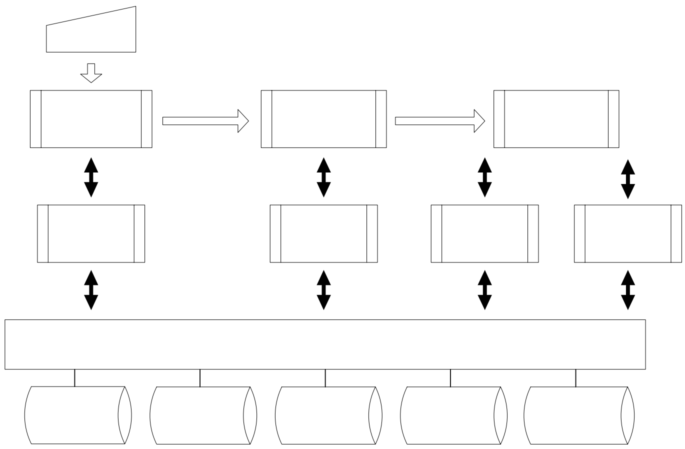

Figure 1 The framework of DIT.

Error Diagnosis

Error Diagnosis relies on the grammatical, lexical, and error type information classified on the knowledge representation platform—refer to figure 1. This information includes error-type definitions compiled from pedagogical expertise and their associated detection procedures. Inflection rules are one example of this error diagnosis procedure. The inflection “ies”, in the word “bussies” above, would trigger the procedure to identify which inflection rule should be applied. Similarly, other common word patterns, programmed into the knowledge map, could invoke associated analysis procedures that, in turn, would detect potential errors or ambiguity.

Interaction via Dialogue

Without grammatical and contextual information, it is difficult to resolve the ambiguity of what the student intends to write. In our DIT model, a clarification dialogue is triggered when ambiguity occurs. For example, we can find two base form candidates in the example above. We encounter the problem of choosing the appropriate inflected form of “bussies”--since it could be either buss or bus depending on which meaning of “bussies” the student wants to use. Therefore, we need to ask the user which one he intends to write. If the student cannot determine which one is suitable in this context, from a tutor’s perspective, we need to give him more information and guide him to correct this sentence. The DIT model also allows the student to ask our system related questions for clarification. Briefly, the interaction mechanism in the DIT model resolves ambiguity and facilitates the teaching process.

Treatment of learners' errors

After the errors are identified, the system will inform the student of the error type and will refer them to related grammatical and lexical information. The dictionary is also useful for generating exercises during the treatment process.

- Exercises are automatically generated from the dictionary's structured knowledge
  Our system provides exercises corresponding to the student's specific mistake. In the case of "bussies", the system will compile other singular nouns that follow the plural inflection rule of "add 'ies'", and administer those to the student as an exercise.
- Suggestions and comments can be retrieved from the knowledge representation platform (InfoMap)
  In addition to general comments given by the system (ex: "bussies": plural error), DIT allows teachers or system administrators to input customized comments about common error types, or about more specific errors.

2.1 Error Classification

DIT identifies errors based on the error classification scheme proposed by Jian (1981) and Ying (1987). Jian divided errors into three categories: lexical errors, grammatical errors and semantic, rhetorical and stylistic errors. Each category can be divided into more specific subcategories. Ying found that student errors are resulted mostly from: overgeneralization of the target language, simplification, and transfer of the student's native language structure onto the target language. This classification scheme is represented in InfoMap. Details for classification and system operation will be illustrated in Section 3.2.

2.2 Lexical Errors Analysis

Lexical errors can be categorized into three classes: compound errors, morphological errors, and spelling errors (Jian, 1981). Examples of compound errors are "well known" (well-known) and "prizefighter" (prize fighter). Morphological errors indicate the learner's miscomprehension about the meaning and function of morphemes and about the morphological rules. The remaining rules are regarded as spelling errors.

To identify compound errors, we pre-process each compound word in the dictionary by the following steps: eliminate the hyphen, combine the two parts into one word, and store it in the compound word table (CWT). If the input word is in the CWT, our system recognizes it as a "compound error".

Morphological errors can be recognized and corrected using the Reduction-Inflection (R-I) algorithm. The basic idea is to reduce the input word to its base form according to several reduction rules, inflect the base form to its inflected forms, and compare the inflected forms with the input. The Reduction-Inflection process is shown in Figure 2.

Reduction → Inflection				
User Input		Base Form		Correct Form
Studys *Studies* *Studyes*	Reduction to base form	*Study (v)*	Inflection according to number or tense	*Studies*

Figure 2 The Reduction-Inflection algorithm applies to "studys", "studies", and "studies".

Figure 3 illustrates the inflection rules for present progressive tense. The details of how to operate R-I algorithm is left to Section 3.2.

Rule 1: Add –ing to the base form of the verb.	e.g. *read → reading; stand → standing*
Rule 2: If a verb ends in a silent –e, drop the final –e and add –ing.	e.g. *leave → leaving; take → taking*
Rule 3: In a one-syllable word, if the last three letters are consonant-vowel-consonant combination, double the last consonant before adding –ing.	e.g. *sit → sitting; run → running*
Rule 3 Exception: Do not double the last consonant in word that end in w, x, or y.	e.g. *sew → sewing; fix → fixing*
Rule 4: In words of two or more syllables that end in a consonant-vowel-consonant combination, double the last consonant only if the last syllable is stressed.	e.g. *admit → admitting; regret → regretting*
Rule 5: If a verb ends in –ie, change the –ie to y before adding –ing.	e.g. *die → dying; lie → lying*

Figure 3 The inflection rules for present progressive tense.

Last, we need to handle other lexical errors. It is established that minimum distance techniques perform well in identification and correction of these errors (Kukich, 1993). Our implementation is based on Wagner's (1974) work.

3. Knowledge Representation

In this section, we introduce our knowledge representation schemes for the DIT model, referred to as the InfoMap.

3.1 InfoMap Basics

InfoMap can be regarded as an ontology that has a tree-like structure. Generally, nodes in InfoMap fall into two categories: concept nodes and function nodes. Concept nodes represent entities, attributes, states, and events; and function nodes show how the concepts are interconnected. The basic function nodes are: Category, Attribute, Instance, Synonym, and Activity. Usually, a root is the name of a domain such as English Tutoring or Elementary School Mathematics Tutoring. Subclass relations organize categories into a taxonomy or taxonomic hierarchy. For example, "morphologic errors" is a subclass of "lexical errors". In addition, InfoMap contains many types of nodes that serve different purposes. Furthermore, many knowledge schemes (such as WORDNET and the Longman Dictionary) can be embedded into InfoMap.

3.2 Using InfoMap to Support the DIT Model

One can imagine that, when examining errors of written English, the examiner (whether student or teacher) needs to manipulate his linguistic knowledge. Our point here is that this knowledge is already represented in InfoMap. DIT exploits this knowledge for two basic functions: detecting lexical errors and offering users useful knowledge concerning the errors. DIT does this through "scripts" that are stored within InfoMap under the relevant nodes of stored knowledge. In triggering these scripts, DIT detects lexical errors and their specific type and also accesses precisely the knowledge relevant for a particular lexical error when it is detected.

Again, the algorithms for identifying specific error types and accessing the relevant information are stored as scripts within InfoMap under a special node-type called "script". How much of this knowledge is appropriate for showing a learner in a particular situation is a pedagogical question. To address this, DIT offers an interface for content designers (domain experts). The interface displays this knowledge that is triggered by specific error types and allows these experts to select from this knowledge exactly the portion that should be shown to the learner. It also allows the designer to decide how to couch this knowledge with appropriate explanations.
One challenge to any ITS is so-called ambiguous errors. It is common when an expert teacher recognizes that a student has made an error, the original intention of the learner is not clear. Therefore, there could be more than one plausible correction. In such cases, it is necessary to recognize the writer's intention so that our system can choose the most appropriate correction. Therefore, we introduce a dialogue mechanism into InfoMap. Under the "Dialogue" node, InfoMap provides several kinds of dialogues to serve different purposes. For example, using the dialogue "Which one do you want to use, A or B?" the content designer could encode his teaching strategy. He could choose to explain A and B individually, and then guide the student to choose what he intended to write using correct the English words.

Once the student's errors are identified, DIT will show the student the error type, related linguistic knowledge (ex. the inflection rule associated with the error), and provide tests, which appear under the "Exercise" function node. Exercise materials can be retrieved from databases or generated from the ontology itself.

We will now describe how InfoMap supports the DIT model. The specific part of InfoMap that supports the DIT model is shown in Figure 4. For reasons of simplicity, general terms are used to describe the scripts associated with all "Script" function nodes. When a student writes: "He lied somewhere which he didn't want to tell," our system first runs the script associated with the "Lexical" node. Within that script, DIT executes the following steps for each word in the input: it looks up the word in the dictionary, checks if the word is spelled incorrectly so that a morphological or compound error occurs. Here, let us focus on explaining how to identify morphological errors with the script associated with the "Morphological Errors" node. The "Morphological Errors" script has three steps. The first step is to reduce the word to its base form by executing the "Reduction" script. The "Reduction" applies the appropriate rule such as "If the word ends in ED drops D," and then checked if the result exists in the Longman Dictionary. If the result is in the dictionary, it will be returned. Take "lied" for example, the "Reduction script" returns "lie". The second step is to inflect the result of the "Reduction script". Our approach is to find its inflected form by looking up the table of irregular verbs. We also apply the appropriate rule to inflect it. In this example, since "lie" has two inflected forms: "lay" and "lied" depending on it's meaning, the DIT model will trigger the following dialogue automatically:

DIT: *If you mean position, use "lay"; if you mean untrue, use "lied." Which one do you intend to use, lay or lied?*

Suppose the student picks “lay”. Since the student’s original input word is “lied”. A morphological error has been identified. According to the student’s specific mistake, DIT generates exercises from the structured knowledge in InfoMap.

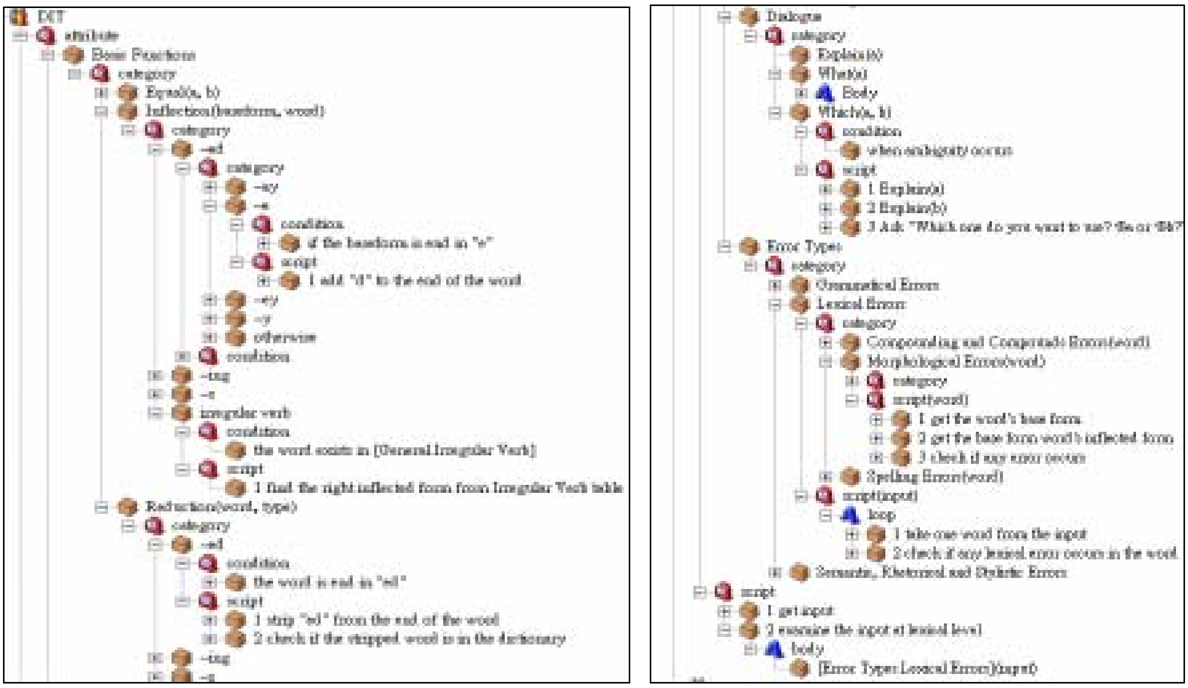

Figure 4 InfoMap supports the DIT model

4. Conclusion

What we hope to have illustrated is that the DIT model, by incorporating a rich and explicit knowledge representation (InfoMap), can address the conceptual level of a learners’ misunderstandings represented by superficial mistakes in the texts they produce. The examples concern morphological errors. Future work on DIT and InfoMap will address grammatical and semantic errors as well and the development of enriched learner modelling based upon user logs that record past student performance.

References

[1] Goodkovsky, V. A. (1996) “*Intelligent Tutor”: Shell, Toolkit & Technology”*. In Proceedings of ITS’96 Workshop on Architectures and Methods for Designing Cost Effective and Reusable ITSs, Montreal, June 10th 1996.

[2] Heffernan, N. T. (2001). *Intelligent tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors*. Ph.D dissertation of the Computer Science Department, Carnegie Mellon University.

[3] Hegarty, C. (1996). *Statutor: Intelligent Tutoring System?*. In Proceedings of BILETA '96 Conference.

[4] Heift, T. & Nicholson, D. (2000). *Theoretical and Practical Considerations for Web-based Intelligent Language Tutoring Systems.* In Proceedings of ITS 2000, Montreal, Canada.

[5] Jian T. H. (1981). *Error Analysis: A study of errors made in written English by Chinese learners*. Master Thesis of English Department, National Taiwan Normal University.

[6] Kukich, K. (1992). *Techniques for Automatically Correcting Words in Text.* ACM Computing Surveys 24(4): 377-439.

[7] Stankov, S. & Božičević J. (1998). *The Computer Tutor in the New Model of Learning and Teaching Control Principles*, In Preprints of the 1st International Workshop: Mechatronics and Industrial Engineering.

[8] Virvou, M., Maras, D., & Tsiriga, V. (2000). *Student Modelling in an Intelligent Tutoring System for the Passive Voice of English Language.* Educational Technology & Society 3(4), 2000.

[9] Wagner, R. A. (1974). *Order-n Correction for Regular Languages*. CACM 17(5): 265-268.

[10] Wu, S. H., Day, M. Y., Tsai, T. H., & Hsu, W. L. (2002). FAQ-centered Organizational Memory. *Knowledge Management and Organizational Memories.* Matta, N. & Dieng-Kuntz, R. (Eds), Kluwer Academic Publishers.

[11] Ying, S. Y. (1987). *Types of Errors in English Compositions by Chinese Students: A search for Interlanguage.* Master Thesis of the Graduate School of Linguistics, Fu Jen Catholic University.