Natural Language Generation for Intelligent Tutoring Systems: a case study

Barbara Di Eugenio [a,1] Davide Fossati [a] Dan Yu [a] Susan Haller [b] Michael Glass [c]
[a] *University of Illinois, Chicago, IL, USA*
[b] *University of Wisconsin Parkside, Kenosha, WI, USA*
[c] *Valparaiso University,Valparaiso, IN, USA*

Abstract. To investigate whether Natural Language feedback improves learning, we developed two different feedback generation engines, that we systematically evaluated in a three way comparison that included the original system as well. We found that the system which intuitively produces the best language does engender the most learning. Specifically, it appears that presenting feedback at a more abstract level is responsible for the improvement.

Keywords. Intelligent Tutoring Systems. Feedback Generation.

1. Introduction

The next generation of Intelligent Tutoring Systems (ITSs) will be able to engage the student in a fluent Natural Language (NL) dialogue. Many researchers are working in that direction [4,6,10,12,14]. However, it is an open question whether the NL interaction between students and an ITS does in fact improve learning, and if yes, what specific features of the NL interaction are responsible for the improvement. From an application point of view, it makes sense to focus on the most effective features of language, since deploying full-fledged dialogue interfaces is complex and costly.

Our work is among the first to show that a NL interaction improves learning. We added Natural Language Generation (NLG) capabilities to an existing ITS. We developed two different feedback generation engines, that we systematically evaluated in a three way comparison that included the original system as well. We focused on aggregation, i.e., on how lengthy information can be grouped and presented as more manageable chunks. We found that syntactic aggregation does not improve learning, but that functional aggregation, i.e. abstraction, does.

We will first discuss DIAG, the ITS shell we are using, and the two NLG systems we developed, *DIAG-NLP1*and *DIAG-NLP2*. Since the latter is based on a corpus study, we will briefly describe that as well. We will then discuss the formal evaluation we conducted and our results.

[1]Correspondence to: B. Di Eugenio, Computer Science (M/C 152), University of Illinois, 851 S. Morgan St., Chicago, IL, 60607, USA. Email: bdieugen@cs.uic.edu.

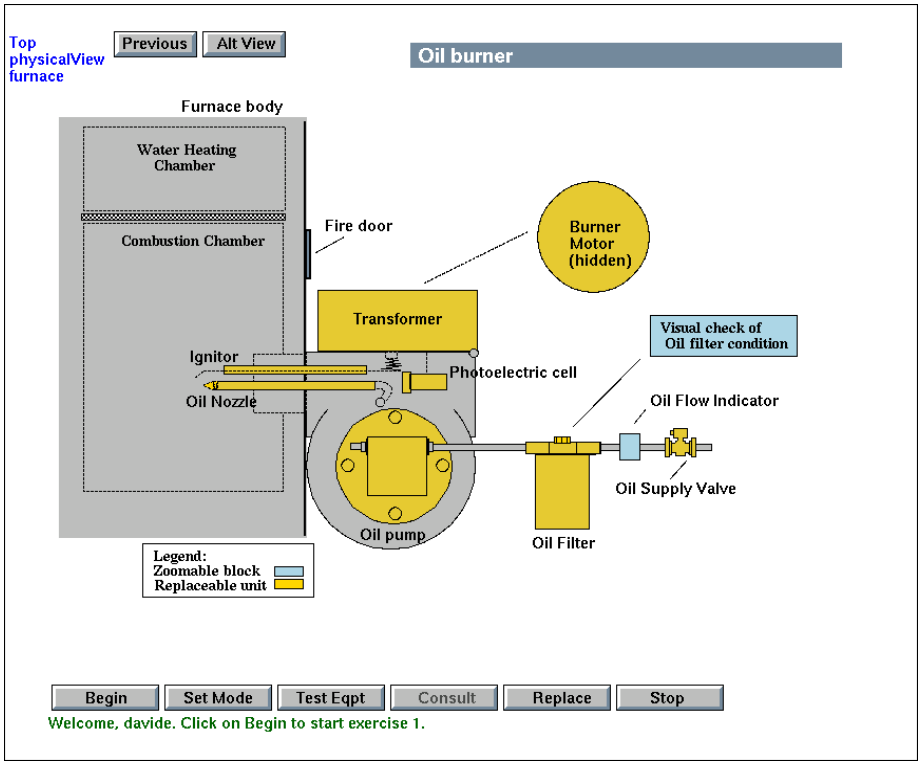

Figure 1. The oil burner

2. Natural Language Generation for DIAG

DIAG [16] is a shell to build ITSs based on interactive graphical models that teach students to troubleshoot complex systems such as home heating and circuitry. DIAG integrates a functional model of the target system and qualitative information about the relations between symptoms and faulty parts (RUs) — RU stands for *replaceable unit*, because the only course of action for a student to fix the problem is to replace RUs in the graphical simulation. A DIAG application presents a student with a series of troubleshooting problems of increasing difficulty. The student tests *indicators* and tries to infer which RU may cause the abnormal states detected via the indicator readings. DIAG's educational philosphy is to push the student to select the most informative tests, and not to provide too much explicit information when asked for hints.

Fig. 1 shows the oil burner, one subsystem of the home heating system in *DIAG-orig*, our DIAG application. Fig. 1 includes indicators such as *Oil Flow indicator*, and many RUs such as *Oil Filter, Ignitor* etc. At any point, the student can consult the tutor via the Consult menu (cf. the Consult button in Fig. 1). There are two main types of queries: *ConsultInd(icator)* and *ConsultRU*. *ConsultInd* queries are used mainly when an indicator shows an abnormal reading, to obtain a hint regarding which RUs may cause the problem. DIAG discusses the RUs that should be most suspected given the symptoms the student has already observed. *ConsultRU* queries are mainly used to obtain feedback on the diagnosis that a certain RU is faulty. DIAG responds with an assessment of that

diagnosis and provides evidence for it in terms of the symptoms that have been observed relative to that RU.

The visual combustion check is igniting which is abnormal (normal is combusting).
Oil Nozzle always produces this abnormality when it fails.
Oil Supply Valve always produces this abnormality when it fails.
Oil pump always produces this abnormality when it fails.
Oil Filter always produces this abnormality when it fails.
System Control Module sometimes produces this abnormality when it fails.
Ignitor Assembly never produces this abnormality when it fails.
Burner Motor always produces this abnormality when it fails.

The visual combustion check indicator is igniting.
This is abnormal.
Normal is combusting.

Within the furnace system,
this is sometimes caused if
the System Control Module has failed.

Within the Oil Burner
this is never caused if
the Ignitor Assembly has failed.
In contrast, this is always caused if
the Burner Motor, Oil Filter, Oil Pump, Oil Supply Valve, or Oil Nozzle has failed.

The combustion is abnormal.
In the oil burner, check the units along the path of the oil and the burner motor.

Figure 2. Answers to *ConsultInd* by *DIAG-orig*, *DIAG-NLP1*and *DIAG-NLP2*

DIAG uses very simple templates to assemble the text to present to the student. As a result, its feedback is highly repetitive and calls for improvements based on NLG techniques. The top parts of Figs. 2 and 3 show the replies provided by *DIAG-orig* to a *ConsultInd* on the *Visual Combustion Check*, and to a *ConsultRu* on the *Water Pump*.

Our goal in developing *DIAG-NLP1* and *DIAG-NLP2* was to assess whether simple, rapidly deployable NLG techniques would lead to measurable improvements in the student's learning. The only way we altered the interaction between student and system is the actual language that is presented in the output window. DIAG provides to *DIAG-NLP1* and *DIAG-NLP2* a file which contains the facts to be communicated – a *fact* is the basic unit of information that underlies each of the clauses in a reply by *DIAG-orig*. Both *DIAG-NLP1* and *DIAG-NLP2* use EXEMPLARS [17], an object-oriented, rule-based generator. EXEMPLARS rules are meant to capture an exemplary way of achieving a communicative goal in a given context.

DIAG-NLP1, which is fully described in [7], (i) introduces syntactic aggregation – i.e., uses syntactic means, such as plurals and ellipsis, to group information [13,15] – and what we call *structural* aggregation, i.e., groups parts according to the structure of the system; (ii) generates some referring expressions; (iii) models a few rhetorical relations (e.g. *in contrast* in Fig. 2); and (iv) improves the format of the output.

The middle part of Fig. 2 shows the output produced by *DIAG-NLP1*(omitted in Fig. 3 because of space constraints). The RUs of interest are grouped by the system modules that contain them (Oil Burner and Furnace System), and by the likelihood that a certain RU causes the observed symptoms. The revised answer highlights that the *Ignitor Assembly* cannot cause the symptom.

Water pump is a very poor suspect.
Some symptoms you have seen conflict with that theory.
Water pump sound was normal.
This normal indication never results when this unit fails.
Visual combustion check was igniting.
This abnormal indication never results when this unit fails.
Burner Motor RMP Gauge was 525.
This normal indication never results when this unit fails.

The water pump is a poor suspect since the water pump sound is ok.
You have seen that the combustion is abnormal.
Check the units along the path of the oil and the electrical devices.

Figure 3. Answers to *ConsultRu* by *DIAG-orig* and *DIAG-NLP2*

2.1. DIAG-NLP2

In the interest of rapid prototyping, *DIAG-NLP1* was implemented without the benefit of a corpus study. *DIAG-NLP2* is the empirically grounded version of the feedback generator. We collected 23 tutoring interactions between a student using the DIAG tutor on home heating and one of two human tutors. This amounts to 272 tutor turns, of which 235 in reply to *ConsultRU* and 37 in reply to *ConsultInd*. The tutor and the student are in different rooms, sharing images of the same DIAG tutoring screen. When the student consults DIAG, the tutor is provided the same "fact file" that DIAG gives to *DIAG-NLP1*and *DIAG-NLP2*, and types a response that substitutes for DIAG's. The tutor is presented with this information because we wanted to uncover empirical evidence for the aggregation rules to be used in our domain.

We developed a coding scheme [5] and annotated the data. We found that tutors provide explicit problem solving directions in 73% of the replies, and evaluate the student's action in 45% of the replies. As expected, they *exclude* much of the information (63% to be precise) that DIAG would provide, and specifically, always exclude any mention of RUs that are not as likely to cause a certain problem, e.g. the *ignitor assembly* in Fig. 2. Tutors do perform a fair amount of aggregation, as measured in terms of the number of RUs and indicators labelled as *summary*. Further, they use functional, not syntactic or structural, aggregation of parts. E.g., the oil nozzle, supply valve, pump, filter, etc., are described as *the path of the oil flow*.

In *DIAG-NLP2* a planning module manipulates the information given to it by DIAG before passing it to EXEMPLARS, and ultimately to RealPro [9], the sentence realizer that produces grammatical English sentences. This module decides which information to include according to the type of query posed to the system. Here we sketch how the reply at the bottom of Fig. 2 is generated. The planner starts by mentioning the referent of the queried indicator and its state (*The combustion is abnormal*), rather than the indicator itself (this is also based on our corpus study). It then chooses, among all the RUs that DIAG would talk about, only those *REL(evant)-RUs* that would definitely result in the observed symptom. It then decides whether to aggregate them functionally by using a simple heuristics. For each RU, its possible aggregators and the number n of units it covers are listed in a table (e.g., *electrical devices* covers 4 RUs, *ignitor, photoelectric cell, transformer* and *burner motor*). If a group of REL-RUs contains k units covered by aggregator *Agg*, if $k < \frac{n}{2}$, *Agg* will not be used; if $\frac{n}{2} \leq k < n$, *Agg* preceded by *some of* will be used; if $k = n$, *Agg* will be used. Finally, *DIAG-NLP2* instructs the student to

check the possibly aggregated REL-RUs.
Full details on the corpus, the coding scheme and DIAG-NLP2 can be found in a companion paper [3].

3. Experimental Results

Our empirical evaluation is a between-subject study with three groups: the first interacts with *DIAG-orig*, the second with *DIAG-NLP1*, the third with *DIAG-NLP2*. The 75 subjects (25 per group) were all science or engineering majors affiliated with our university. Each subject read some short material about home heating, went through one trial problem, then continued through the curriculum on his/her own. The curriculum consisted of three problems of increasing difficulty. As there was no time limit, every student solved every problem. Reading materials and curriculum were identical in the three conditions.

While a subject was interacting with the system, a log was collected including, for each problem: whether the problem was solved; total time, and time spent reading feedback; how many and which indicators and RUs the subject consults DIAG about; how many, and which RUs the subject replaces. We will refer to all the measures that were automatically collected as *performance measures*.

At the end of the experiment, each subject was administered a post-test, a test of whether subjects remember their actions, and a usability questionnaire.

We found that subjects who used *DIAG-NLP2* had significantly higher scores on the post-test, and were significantly more correct in remembering what they did. As regards performance measures, there are no so clear cut results. As regards usability, subjects prefer the NL enhanced systems to *DIAG-orig*, however results are mixed as regards which of the two they actually prefer.

In the tables that follow, boldface indicates significant differences, as determined by an analysis of variance performed via ANOVA, followed by post-hoc Tukey's tests.

	Post-Test	RU Precision	RU Recall
DIAG-orig	0.72	0.78	0.53
DIAG-NLP1	0.69	0.70	0.47
DIAG-NLP2	**0.90**	**0.91**	0.40

Table 1. Learning Scores

Table 1 reports learning measures, average across the three problems. The post-test consists of three questions and tests what the student has learnt about the domain. Subjects are also asked to remember the RUs they replaced, under the assumption that the better they remember how they solved a certain problem, the better they will be able to apply what they learnt to a new problem - namely, their recollection should correlate with *transfer*. We quantify the subjects' recollec-

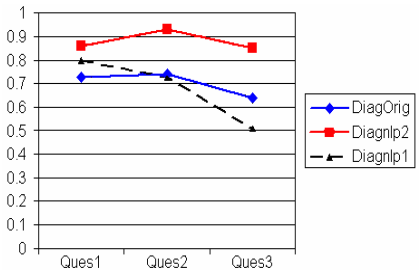

Figure 4.Scores on problems

tions in terms of precision and recall with respect to the log that the system collects. *DIAG-NLP2* is significantly better as regards post-test score ($F = 10.359, p = 0.000$), and RU Precision ($F = 4.719, p = 0.012$).

Performance on individual questions in the post-test is illustrated in Fig. 4. Scores in *DIAG-NLP2* are always higher, significantly so on questions 2 and 3 ($F = 8.481, p = 0.000$, and $F = 7.909, p = 0.001$), and marginally so on question 1 ($F = 2.774, p = 0.069$).

	Time	RU Replaced	ConsultInd	Avg. Time	ConsultRU	Avg. Time
DIAG-Orig	30'17"	8.88	22.16	8"	63.52	5"
DIAG-NLP1	28'34"	11.12	**6.92**	14"	45.68	4"
DIAG-NLP2	34'53"	11.36	28.16	**2"**	52.12	5"

Table 2. Performance Measures across the three systems

Table 2 reports performance measures, cumulative across the three problems (other than average reading times, *Avg. Time*). Subjects don't differ significantly in the time they spend solving the problems, or in the number of RUs they replace, although they replace fewer parts in *DIAG-orig*. This trend is opposite what we would have hoped for, since when repairing a real system, replacing parts that are working should clearly be kept to a minimum. The simulation though allows subjects to replace as many as they want without any penalty before they come to the correct solution.

The next four entries in Table 2 report the number of queries that subjects ask, and the average time it takes subjects to read feedback from the system. The subjects ask significantly fewer *ConsultInd* in *DIAG-NLP1* ($F = 8.905, p = 0.000$), and take significantly less time reading *ConsultInd* feedback in *DIAG-NLP2* ($F = 15.266, p = 0.000$). The latter result is not surprising, since the feedback in *DIAG-NLP2* is in general much shorter than in *DIAG-orig* and *DIAG-NLP1*. Neither the reason not the significance of subjects asking fewer *ConsultInd* of *DIAG-NLP1* are apparent to us.

We also collected usability measures. Although these are not usually reported in ITS evaluations, in a real setting students should be more willing to sit down with a system that they perceive as more friendly and usable. Subjects rate the system along four dimensions on a five point scale: clarity, usefulness, repetitiveness, and whether it ever misled them (the highest clarity but the lowest repetitiveness receive 5 points). There are no significant differences on individual dimensions. Cumulatively, *DIAG-NLP2* (at 15.08) slightly outperforms the other two (*DIAG-orig* at 14.68 and *DIAG-NLP1* at 14.32), however, the difference is not significant (highest possible rating is 20 points). Finally, on paper, subjects compare two pairs of versions of feedback: in each pair, the first feedback is generated by the system they just worked with, the second is generated by one of the other two systems. Subjects say which version they prefer, and why (they can judge the system along one or more of four dimensions: natural, concise, clear, contentful). In general, subjects prefer the NLP systems to *DIAG-orig* (marginally significant, $\chi^2 = 9.49, p < 0.1$). Subjects find *DIAG-NLP2* more natural, but *DIAG-NLP1* more contentful ($\chi^2 = 10.66, p < 0.025$).[1]

[1] In these last two cases, χ^2 is run on tables containing the number of preferences assigned to each system, in the various categories.

4. Discussion and future work

Only very recently have the first few results become available, to show that first of all, students do learn when interacting in NL with an ITS [6,10,12,14]. However, there are very few studies like ours, that compare versions of the same ITS that differ in specific features of the NL interface. One such study is [10], which found no difference in the learning gains of students who interact with an ITS that tutors in mechanics using typed text or speech.

We did find that different features of the NL feedback impact learning. We claim that the effect is due to using functional aggregation, that stresses an abstract and more conceptual view of the relation between symptoms and faulty parts. However, the feedback in *DIAG-NLP2* changed along two other dimensions: using referents of indicators instead of indicators, and being more strongly directive in suggesting what to do next. Although we introduced the latter in order to model our tutors, it has been shown that students learn best when prompted to draw conclusions by themselves, not when told what those conclusions should be [2]. Thus we would not expect this feature to be responsible for learning.

Naturally, DIAG-NLP2 is still not equivalent to a human tutor. Unfortunately, when we collected our naturalistic data, we did not have students take the post-test. However, performance measures were automatically collected, and they are reported in Table 3 (as in Table 2, measures other than reading times are cumulative across the three problems). If we compare Tables 2 and 3, it is apparent that when interacting with a human tutor,

Time	RU Replaced	ConsultInd	Avg. Time	ConsultRu	Avg. Time
38'54"	8.1	1.41	21.0"	10.14	14.0"

Table 3. Performance Measures when interacting with human tutors

students ask far fewer questions, and they read them much more carefully. The replies from the tutor must certainly be better, also because they can freely refer to previous replies; instead, the dialogue context is just barely taken into account in *DIAG-NLP2* and not taken into account at all in *DIAG-orig* and *DIAG-NLP1*. Alternatively, or in addition, this may be due to the *face* factor [1,11], i.e., one's public self-image: e.g., we observed that some subjects when interacting with any of the systems simply ask for hints on every RU without any real attempt to solve the problem, whereas when interacting with a human tutor they want to show they are trying (relatively) hard. Finally, it has been observed that students don't read the output of instructional systems [8].

The DIAG project has come to a close. We are satisfied that we demonstrated that even not overly sophisticated NL feedback can make a difference; however, the fact that *DIAG-NLP2* has the best language and engenders the most learning prompts us to explore more complex language interactions. We are pursuing new exciting directions in a new domain, that of introductory Computer Science, i.e., of basic data structures and algorithms.

Acknowledgments. This work is supported by grants N00014-99-1-0930 and N00014-00-1-0640 from the Office of Naval Research. We are grateful to CoGenTex Inc. for making EXEMPLARS and RealPro available to us.

References

[1] Penelope Brown and Stephen Levinson. *Politeness: Some Universals in Language Usage*. Studies in Interactional Sociolinguistics. Cambridge University Press, 1987.

[2] Michelene T. H. Chi, Stephanie A. Siler, Takashi Yamauchi, and Robert G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25:471–533, 2001.

[3] B. Di Eugenio, D. Fossati, D. Yu, S. Haller, and M. Glass. Aggregation improves learning: experiments in natural language generation for intelligent tutoring systems. In *ACL05, Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, 2005.

[4] M. W. Evens, J. Spitkovsky, P. Boyle, J. A. Michael, and A. A. Rovick. Synthesizing tutorial dialogues. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 137–140, Hillsdale, New Jersey, 1993. Lawrence Erlbaum Associates.

[5] M. Glass, H. Raval, B. Di Eugenio, and M. Traat. The DIAG-NLP dialogues: coding manual. Technical Report UIC-CS 02-03, University of Illinois - Chicago, 2002.

[6] A.C. Graesser, N. Person, Z. Lu, M.G. Jeon, and B. McDaniel. Learning while holding a conversation with a computer. In L. PytlikZillig, M. Bodvarsson, and R. Brunin, editors, *Technology-based education: Bringing researchers and practitioners together*. Information Age Publishing, 2005.

[7] Susan Haller and Barbara Di Eugenio. Minimal text structuring to improve the generation of feedback in intelligent tutoring systems. In *FLAIRS 2003, the 16th International Florida AI Research Symposium*, St. Augustine, FL, May 2003.

[8] Trude Heift. Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? *ReCALL Journal*, 13(2):129–142, 2001.

[9] Benoît Lavoie and Owen Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.

[10] D. J. Litman, C. P. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman. Spoken versus typed human and computer dialogue tutoring. In *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*, 2004.

[11] Johanna D. Moore, Kaska Porayska-Pomsta, Sebastian Varges, and Claus Zinn. Generating tutorial feedback with affect. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, 2004.

[12] S. Peters, E. Owen Bratt, B. Clark, H. Pon-Barry, and K. Schultz. Intelligent systems for training damage control assistants. In *Proceedings of I/ITSEC 2004, Interservice/Industry Training, Simulation, and Education Conference*, 2004.

[13] Mike Reape and Chris Mellish. Just what *is* aggregation anyway? In *Proceedings of the European Workshop on Natural Language Generation*, Toulouse, France, 1998.

[14] C. P. Rosé, D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. Exploring the effectiveness of knowledge construction dialogues. In *AIED03, Proceedings of AI in Education*, 2003.

[15] James Shaw. A corpus-based analysis for the ordering of clause aggregation operators. In *COLING02, Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[16] Douglas M. Towne. Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*, 1997.

[17] Michael White and Ted Caldwell. Exemplars: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 266–275, 1998.