Deriving content selection rules from a corpus of non-naturally occurring documents for a novel NLG application

Conference or Workshop Item

Deriving content selection rules from a corpus of non-naturally occurring documents for a novel NLG application

Sandra Williams and Ehud Reiter
University of Aberdeen
Computing Science Department, Aberdeen, AB24 3UE, U.K.
{ swilliam, ereiter }@csd.abdn.ac.uk

Abstract

We describe a methodology for deriving content selection rules for NLG applications that aim to replace oral communications from human experts by written communications that are generated automatically. We argue for greater involvement of users and for a strategy for handling sparse data.

1 Introduction

One of the challenges in building NLG systems is to derive content selection (CS) rules, typically by combining corpus knowledge and domain (and user) knowledge. CS rules specify what information the output document should communicate for a particular set of input data. For instance, they often map input data to fragments of document content.

Novel Natural Language Generation (NLG) applications produce documents that do not occur naturally. A subset of these applications communicate information *in written form* that humans would normally deliver *orally*. Such applications pose particular problems for NLG developers as we demonstrate in this paper by using our own development of the SkillSum system as a case study. More specifically we describe the part of SkillSum's development that involved the derivation of content selection (CS) rules.

SkillSum [Williams and Reiter, 2005] generates personalised basic skills feedback reports. Basic-skills tutors usually give feedback orally, but SkillSum attempts to generate a written report that communicates similar information to an adult student who has just completed a basic skills test. STOP [Reiter *et al*., 2003b] was another NLG system that attempted to communicate information that was normally delivered orally. STOP generated personalised smoking cessation letters, which communicated the kind of advice that would normally be given orally by a GP during a personal one-to-one consultation.

Tutors and doctors give their feedback orally partially because information about skills and health can be very sensitive and personal, and partially because discussing the topic orally is quicker (and hence cheaper) than writing a written report. They also recognize that there are advantages to written texts (notably that the recipient can take a written text home and think about it, and also discuss it with family and friends), and in general would like to have the option of giving people written texts, if these could be produced quickly and cheaply. Hence our interest to using NLG to automatically produce such texts.

The problem of deriving CS rules from corpora and domain/user knowledge is very hard, but is often glossed over; it is assumed that we can magically come up with successful CS rules even when there is little evidence for what "good content" might be for a given application or whether the content will be right for users. We can derive CS rules from corpora if we have a large parallel corpus of input data and manually-authored output texts, which covers most permutations of inputs and outputs (e.g. the parallel corpus of the SumTime system [Sripada *et al*. 2003]). Another situation where it might be relatively easy to derive CS rules is if the specification of what content should be present in the output is well defined and the application requires only a small, simple set of content and message types to be generated. In such cases, the methodologies presented by Reiter and Dale (2000) and Geldof (2003) can be used. However, none of these was the case with SkillSum (or in STOP).

In this paper we describe how a corpus of expert-authored basic skills reports was collected and analysed. We focus on the problem of CS rule derivation and in particular on problems we encountered with variability of content and sparsity of data in the corpus and in trying to incorporate the requirements of users which sometimes conflicted with experts' advice.

2 Related Work

Our starting point for determination of content selection rules was Reiter and Dale [2000], Geldof [2003] and Reiter and Sripada [2002]. Reiter and Dale describe a method for deriving knowledge from a corpus and Geldof essentially extends the method for message types, referring expressions, aggregation types and lexical choice. We followed Geldof's method to derive high-level document structure, but at lower-levels, the content of SkillSum's feedback reports is far less clear-cut than that of Geldof's route descriptions. There are fewer obligatory elements and there is not the same rigid logical ordering as that imposed by physical routes. We also incorporated many of the KA techniques discussed by Reiter *et al*. [2003a]. These methods work best when there are a few expert-authored texts for every possible set of system inputs. Since we did not have this, we extended the methodology to include extrapolation of rules to cover missing corpus data and allowed users to have a say in content selection. This follows Schneiderman's advice [2000] to accommodate users, even though they are not normally involved at the content selection stage.

SkillSum differs from existing NLG applications in that the content of its output texts serves very different communicative purposes from those of many existing NLG applications. In fact basic skills reports include multiple kinds of communicative purpose. Other systems generate descriptions, or explanations, or instructions, or advice. But SkillSum's content is complex in that it includes many of these: descriptions, interpretations, advice *and* instructions. SkillSum's content is also different from tutoring systems that generate explanations and instructions (e.g. [Moore *et al.*, 2004]) and from Cogentex's Recommender system that generates camera purchasing advice (see www.cogentex.com/solutions/recommender/). From this perspective it is perhaps most similar to STOP [Reiter *et al*., 2003b].

A related text type to basic skills summary reports is school reports. However, Education literature on writing school reports is unhelpful because adult basic learners have often had bad experiences with school and school reports in the past [FENTO, 2004].

3 The SkillSum Application

SkillSum is a web-based application for basic skills testing and feedback report generation. Users of SkillSum can test either their literacy or numeracy in a short test consisting of a maximum of 27 questions and then receive a personalised report automatically generated using NLG technology (see Figure 1 for example output).

The intended users of SkillSum are adults aged 16 years and over with low basic skills, but not with severe learning difficulties. Such adults occur in large numbers in the UK; up to one fifth of the adult population, according to a Government survey [Moser, 1999]. SkillSum is a collaborative project between a commercial partner, Cambridge Training and Development Ltd. (CTAD) and researchers at Aberdeen University. CTAD developed SkillSum's basic skills testing module and Aberdeen the feedback report generator.

SkillSum's basic skills testing component originally contained much longer tests from which diagnostics were possible. Trials with users revealed that these tests would take too long to complete and users would need support whilst doing them. We switched to shorter tests that could be completed with little support. The shorter tests are "screeners", that is, they identify problems with basic skills but do not include diagnostics. This switch had quite an impact on feedback report content. For instance, the diagnostic part of reports became more general and vague.

Initially SkillSum was to be used at home or in Internet access centres. Now SkillSum is to be used in further education colleges where all incoming students are normally screened for basic skills problems so that help with basic English and Maths can be provided. Again, the change had an impact on feedback report content. Now users are college students who have just (or are just about to) enrol in a course. We hypothesised that such students would want to know if their skills were adequate for their intended course. Trials with students confirmed that this type of content was relevant.

We have run trials of SkillSum prototypes in a number of colleges. To date, we have developed five SkillSum prototypes and have carried out trials of each one in a continuous cycle of system development, testing, user trials and improvement.

4 Deriving CS Rules for SkillSum

In SkillSum, since the output texts did not occur naturally, we asked human experts to write some examples for us. This produced a fairly small corpus (Section 4.3), which suffered from data sparsity (i.e. corpus texts covered only a small fraction of the possible permutations of inputs to the system). We therefore found it necessary to manually extrapolate our CS rules to account for cases not covered by the corpus.

Reiter and Sripada (2002) found that corpora of expert-written texts cannot always be considered as gold standards for NLG because experts disagree and they can make mistakes. In SkillSum too, it was clear that our expert-authored corpus of sample basic skills feedback reports could not necessarily be regarded as the last word on what the content should be because our experts differed in their choice of content and users disagreed with experts.

We encountered another problem with SkillSum. Because our application was novel, we did not have a clear specification of what content should be included at the start of the project. In fact our specification developed gradually over the first year of the project. Other novel NLG applications may also suffer from this problem.

Reiter and Dale (2000) say that *"the goal of many NLG systems is to produce documents which are as similar as possible to documents produced by human experts*". Our goal was not just to mimic the content chosen by experts, but also to consider the requirements of users and to make the content useful and relevant to them. We therefore listened to users too and came up with CS rules for content that we hoped would be relevant and motivating for them as well as being acceptable to basic skills experts. Allowing users to have their say in the final specification of content is a departure from previous NLG CS rule derivation methodologies.

Our methodology was to work closely with both domain experts and with the intended users of the system and to combine knowledge acquired from them with knowledge from the corpus. This paper describes how we derived CS rules from a semi-automatically analysed corpus of human authored example output texts. Indeed, in some cases, as we will demonstrate in this paper, user requirements and extrapolation of rules played a more important role than corpus data.

In some NLG applications, if the content is wrong the cost might not be all that high. For example, irrelevant information could simply be ignored by users. But in SkillSum if the content is wrong, the cost is high; users can get angry, and (even worse) they may lose interest in improving their skills. Hence our development methodology stresses regularly producing and evaluating prototype systems, and using feedback from real users to improve document content.

4.1 What is in a basic skills summary report?

SkillSum reports should help people understand their basic skills strengths and weaknesses and advise them (if necessary) on how to get help. Although experts agreed that this type of content should be present, it still posed a challenge as to what to say exactly, because the topic is a very sensitive one indeed. Telling people with low self-confidence that they have problems with their literacy and/or numeracy can be hurtful! As mentioned above, the cost of getting the content wrong could be very high indeed.

Some issues that we considered include:

- Should reports mention students' mistakes as well as their correct answers?
- Should reports congratulate students for doing well, when perhaps their performance was worse than normal, or, on the other hand, should they commiserate with students when perhaps their performance was better than normal?
- Should reports attempt to motivate users by referring to their ambitions (e.g. qualifications or career) and how much (if any) motivational content should be included?
- How much advice should be given?

Experts tend to agree that reports should be encouraging, focus on positive aspects and not mention mistakes. On the other hand, the users often wanted to know what their mistakes were (it can be frustrating to score twenty-six out of twenty-seven and not know which one was wrong!). To address this, current reports include "more information" links to pop up a list of questions with correct and incorrect answers.

Not knowing an individual and how much effort he/she has put into the test remains an unsolved problem. Inclusion of evaluative comments such as *"this is very good"* is meaningless without such knowledge and, indeed they are meaningless without reference to some scale (but experts advised against mentioning the U.K. basic skills core curriculum scale as students are not familiar with it).

Another difficulty was what to say to people who could not answer many, or any, of the screener questions correctly. Also we cannot tell if this was because something went wrong during the test (since it is web-based): perhaps there was a problem with using the computer or the mouse; or perhaps the user has severe learning difficulties. Current reports for these users are very short indeed and advise talking to a tutor.

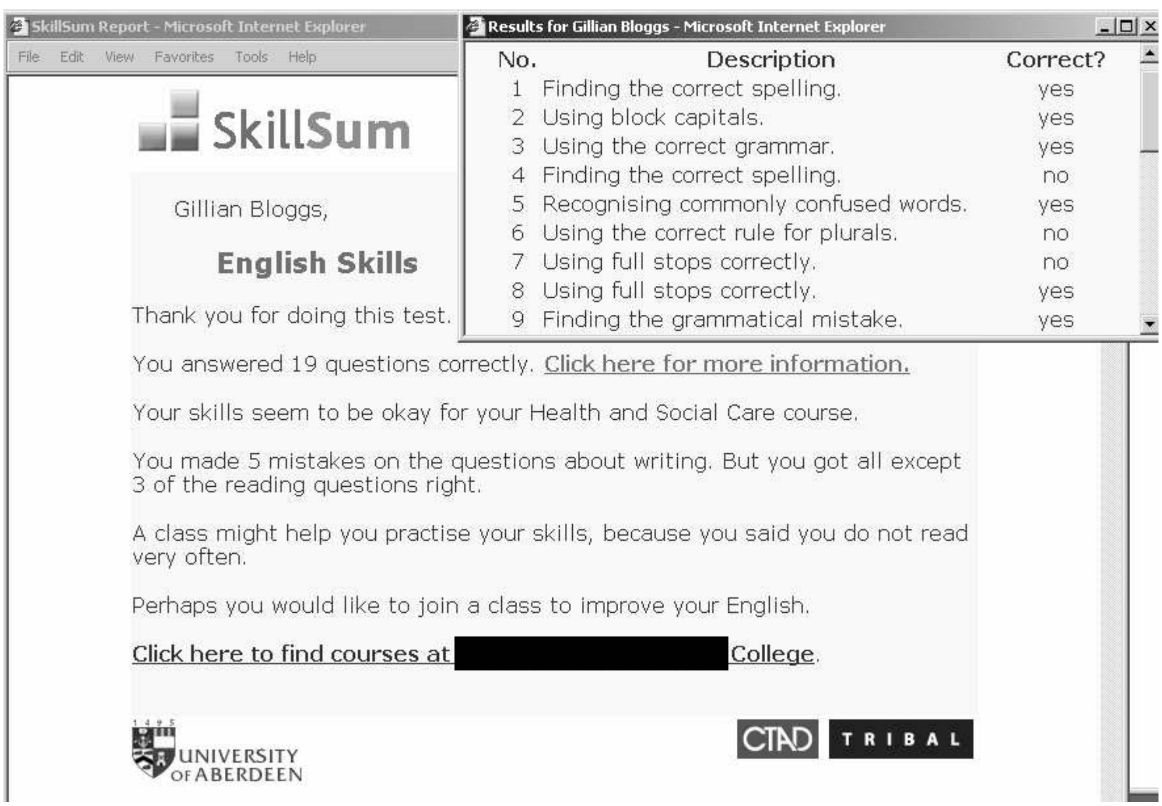

Figure 1. Output from SkillSum showing pop-up with "more information".

A student MSc project [Tintarev 2004], experimented with generating motivational content based on a user's intrinsic motivations (e.g. self-confidence and self-esteem) but an evaluation showed that it was not very successful because it is very difficult to obtain this data simply and reliably from users. This could have potential for future work but for the present, we confine SkillSum's motivational content to encouraging a user to gain a sufficient standard in basic skills for his/her course (since data about courses can be obtained simply and reliably).

4.2 Baseline System

CTAD's software currently generates very simple reports, which just give a score and overall literacy level, and suggest discussing this with a tutor. For example:

Andy Jones
Date: 06/10/2004
Thank you for doing this test. You scored 13. You may need help with level 1 literacy. Talk to your tutor or supervisor."

This is the baseline from which to measure NLG output.

4.3 KA and corpus collection

At the beginning of the project we collected a corpus of expert-authored texts relating to the longer basic skills test. For the new screener tests we elicited new expert-authored reports for nine case studies in literacy and nine in numeracy. An alternative method would have been to record tutor-student feedback sessions, but due to the sensitive and confidential nature of these, we preferred a method that was less intrusive; we also wanted the expert tutors to consider the issues involved in producing written reports. We gave them test results and short user profiles containing background material, e.g. age, current course and/or job wanted and self-assessments of maths and English skills (these were built using anonymous data from actual people who took part in earlier pilots and from [Swain *et al.*, 2004]). Two such profiles can be seen in the Appendix along with two example reports written by an expert.

We also acquired 1500 sets of test results (that is, input data for SkillSum); this gave us an idea of the range of inputs that SkillSum needed to cover.

4.4 Deriving high level document structure

An analysis of the tutor-authored reports demonstrated that they have similarities in high-level content structures but individual author differences in lower-level content. Our high-level analysis essentially followed the methodology of Geldof [2003]. Most of the expert-authored basic skills feedback reports included an initial section, a description of results (summary) an interpretation of the results (diagnosis), advice on what to do next (advice section) and a site-tailored link. For example, the report shown in Figure 1 has the following high-level structure.

- **Initial**:*"Gillian Bloggs, English Skills Thank you for doing this test."*
- **Summary**: *"You answered 19 questions correctly. Click here for more information."* Link to pop-up list of questions.
- **Diagnosis**: *"Your skills seem to be okay for your Health and Social Care course. You made 5 mistakes on the questions about writing. But you got all except 3 of the reading questions right."*
- **Advice**: *"A class might help you practise your skills, because you said you do not read very often. Perhaps you would like to join a class to improve your English."*
- **Site-tailored link**: *"Click here to find courses at Peterborough Regional College."*

Notice that even though Gillian's skills were considered good enough for her course, she is still being encouraged to take a class to improve. This is because her level of English is not very high.

Most expert-authored reports follow a similar basic structure. Sometimes "thanks" in the initial section was not present. The summary section was always present. The diagnosis section was not present in reports for students who had answered less than five questions correctly. As the overall score increased, the length of diagnosis and advice increased and sometimes these more complex sections were interleaved "diagnosis, advice, diagnosis, advice" and so on.

As the high-level structure emerged, it became clear what types of user knowledge we need to elicit in order to generate reports, e.g. self-assessments of skills, frequencies of reading, writing and calculations and information about users' courses.

4.5 Lower-level content

Lower-level content was analysed using a number of methods. We parsed the entire expert-authored corpus using MIT's parser (web.media.mit.edu/~hugo/montylingua/) and identified Message Types as Geldof suggests [2003]. The parsed corpus was used to model deep syntactic structures.

To derive rules for lower-level content, we attempted to draw flowcharts with decision points for individual messages inclusion, but we found the most successful method for SkillSum turned out to be manual RST analysis [Mann and Thompson 1998] of the entire corpus to give a number of RST tree fragment structures, e.g. one per piece of diagnosis or advice, like that shown in Figure 2.

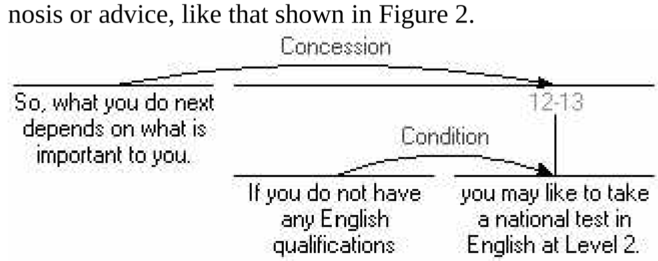

Figure 2. RST analysis of a piece of Advice (strings represent deep syntax).

This was followed by manual simplification (e.g. removal of cue phrases and concession satellite in Figure 2) and then construction of individual rules for inclusion of the templates (based on the user's results in the Screener and the user model). The document planning module outputs these RST structures which are then input to the microplanner that makes choices on ordering, cue phrases aggregation and punctuation, see Williams [2004].

4.6 Extrapolating CS rules to cover missing corpus data

The corpus was small so we extrapolated from existing data. In practice, CS rules are expressed as if-then rules and sometimes, for instance, the "else" part might be missing. Extrapolating the rule would mean supplying the missing part. For example, *"you should have the reading skills to be able to cope with your sports course"* was present (one occurrence) in the corpus, but there was no data about what to say when skills were inadequate for the user's course. The rule for including this content was therefore extrapolated by adding an "else" part to the rule as follows:

IF *the user is about to begin a Level 1 course at college*
  AND *his/her English skills are at least Level 1,*
THEN
    Add content to advise the user that his/her skills are adequate for his/her course.
ELSE*,*
    Add content to advise the user that his/her skills are inadequate
    IF *he/she is not already receiving help with basic skills,*
    THEN *add content that he/she should try to improve his/her skills, e.g. by taking a basic skills course.*

Of course we piloted extrapolated rules with experts and users. The above rule turned out to be one of the most important CS rules in SkillSum (in the sense that it is currently deployed in the generation of every report where the user's course is known) although only part of it actually occurred in our corpus.

4.7 Incorporating domain knowledge and users' comments

The above rule incorporates domain knowledge about courses and levels, knowledge about the user (e.g. what course the user is about to take and whether he/she is receiving help with basic skills) and expert knowledge (advise the student to take up a basic skills course).

We also incorporated comments from users. For example, many users told us they wanted to know which questions they got right and wrong, so we added this information in a pop-up window (see Figure 1).

4.8 CS Rule Derivation Methodology in SkillSum

Current practice in the NLG community (at least as described in published papers) is to base CS rules on KA with experts and corpus analysis. The methodology we used in SkillSum also includes rule extrapolation and the involvement of end users. The following list summarises our method which we also illustrate in Figure 3.

Knowledge Acquisition:

- Interview experts to elicit domain knowledge.
- Collect sets of results and sets of personal details (user models).
- Ask experts to write example reports (corpus of output texts)

System Development:

- Derive rules for analysing existing input data.
- Determine what additional input data (if any) is needed (such as background information about the user).

.

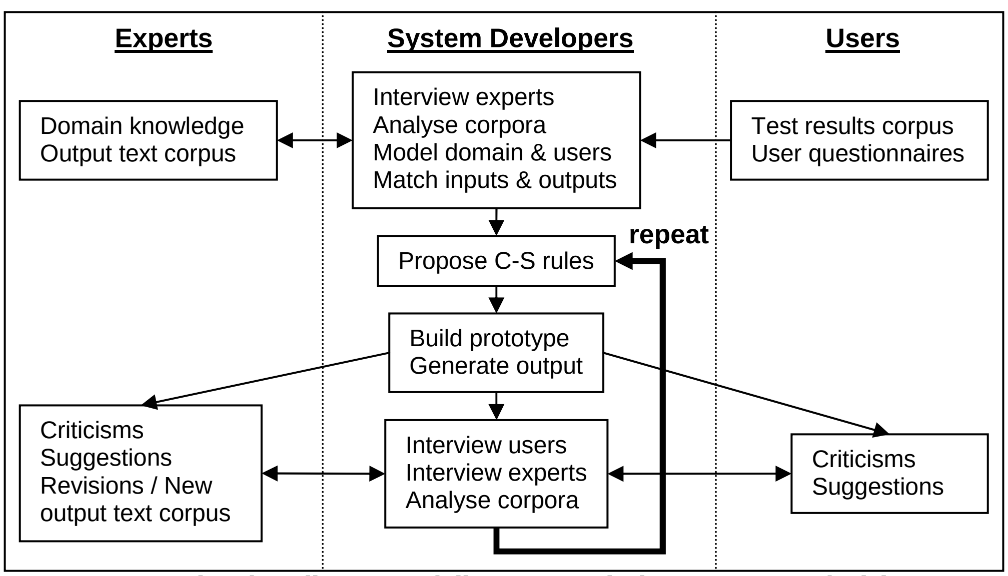

Figure 3. Flowchart illustrating SkillSum's CS rule determination methodology

- Derive content and message types from corpus texts, and use RST analysis.
- Derive rules for selecting content from the input data using available domain and user knowledge **and extrapolate**.

System Trial:

- Ask experts to criticise content of generated reports, edit them and write alternative reports.
- Pilot with users and ask them to criticise and suggest improvements.

The flowchart in Figure 3 shows how we incorporate knowledge from experts (on the left-hand side) and from users (on the right-hand side) when developing CS rules. The central system developer section shows how we repeat the last three stages a number of times. To date with SkillSum we have repeated these steps five times. At a recent trial of SkillSum, 13 out of 15 students preferred the content of the NLG output over that of the baseline system described in Section 4.2 (significant at $p<0.01$ in a binomial test).

5 Conclusions

CS rule derivation is a complex process. In this paper we have described our methodology for deriving CS rules for SkillSum, one of a class of novel NLG applications where output that is normally delivered orally by humans is generated by the NLG system in written form. We have argued that the methodology should include a strategy for handling sparse data when the corpus of expert-authored texts is small and also that users should have a chance say what they find useful and relevant in terms of document content. We feel that users can make a valuable contribution to the process of CS rule derivation. We hope that developers of similar NLG systems will find our experiences with SkillSum useful.

Acknowledgments

We thank the reviewers, CTAD and members of the Aberdeen NLG group for their help and comments. This research is supported by U.K. PACCIT-LINK Grant ESRC RES-328-25-0026.

References

[FENTO, 2004] Further Education National Training Organisation (FENTO). Including Language, Literacy and Numeracy Learning in all Post-16 Education. www.nrdc.org.uk.

[Geldof, 2003] Geldof, S. Corpus analysis for NLG. In: Reiter, E., Horacek, H. and van Deemter, K. (Eds.) *9th European Workshop on NLG*, 31-38.

[Mann and Thompson, 1987] W. Mann and S. Thompson. Rhetorical Structure Theory: A Theory of Text Organization. ISI, Reprint Series no. ISI/RS-87-190.

[Moore *et al.*, 2004] Johanna D. Moore, Kaska Porayska-Pomsta, Sebastian Varges, and Claus Zinn. Generating Tutorial Feedback with Affect. *Proc.17th International FLAIRS Conference*, AAAI Press.

[Moser 1999] Moser, C. Improving literacy and numeracy: a fresh start. Report of the working group chaired by Sir Clause Moser. www.lifelonglearning.co.uk/mosergroup/

[Reiter and Dale, 2000] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. CUP.

[Reiter and Sripada, 2002] Ehud Reiter and Somayajulu Sripada. Should Corpora Texts be Gold Standards for NLG? *Proceedings of INLG02*, 97-104.

[Reiter *et al.*, 2003a] Ehud Reiter, Somayajulu Sripada and Roma Robertson. Acquiring Correct Knowledge for Natural Language Generation, *Journal of Artificial Intelligence Research*, 18:491-516.

[Reiter *et al.*, 2003b] Ehud Reiter, Roma Robertson, and Liesl Osman. Lessons from a Failure: Generating Tailored Smoking Cessation Letters. *Artificial Intelligence*, 44:41-58.

[Schneiderman, 2000] Ben Schneiderman. Universal Usability: Pushing human-computer interaction research to empower every citizen. Communications of the ACM, 43, 84-91.

[Sripada *et al*., 2003] S. G. Sripada, Ehud Reiter, Jim Hunter and Jin Yu. Exploiting a parallel TEXT-DATA corpus. *Proceedings of Corpus Linguistics*, 734-743.

[Swain *et al.*, 2004] Swain, Jon., Elizabeth Baker, Debbie Holder, Barbara Newmarch and Diana Coben. Beyond the daily application: Making numeracy teaching meaningful to adult learners. NRDC. www.nrdc.org.uk.

[Tintarev, 2004] Nava Tintarev. Content Determination for Reports Aimed at Adult Literacy Learners. MSc Thesis, Uppsala Universitet.

[Williams and Reiter, 2005] Sandra Williams and Ehud Reiter. Generating readable texts for readers with low basic skills. To appear in *Proceedings of the 10th European Workshop on NLG*.

[Williams, 2004] Williams, Sandra, H. *Natural Language Generation of discourse relations for different reading levels.* PhD Thesis, University of Aberdeen, 2004.

Appendix 1. Extracts from Experts' Questionnaire

Instructions for authors

We would like you to write (or edit) a short feedback report for each of the people described in the following learner profiles. Please try to use everything in the profile that you consider relevant. You can write anything you think appropriate, as if the people in the profiles had come to you personally as a consultant with their screener results. They are asking for feedback and advice and you want to encourage them as much as possible. Assume that you know everything that the profile says about them. It might help to imagine the person is sitting next to you and that you are talking to him/her.

What to include

- The report should tell the person how well they have done in the screener. We have given you their screener results and a list of the questions in each screener.
- If you think it is relevant, you could mention right or wrong answers to particular questions and/or particular skills that you think that the person has mastered, or has not yet been taught. How would you explain to this person what he/she is good at and/or what he/she still needs to learn?
- You want to encourage the person to improve his/her skills. Can you write something that he/she would find motivating?
- Please also tell us which parts of the "character and background" are useful.

Restrictions

- Reports for people with the lowest scores should be very short indeed (say, 50 words), but they still ought to contain some comments and advice that the person would find useful. Reports for people with higher scores can be a little longer.
- A recent CTAD strategy meeting decided, I think, that reports should not mention levels or the curriculum, so you ought not to mention those.

Learner Profile #2 (to be edited): Literacy Screener

Name: John Smith **Age:** 18 **His own assessment of his Reading and Writing:** "okay"
Overall Screener Score: 8 **Time taken**: 8 mins.
Background (Stoke-on-Trent)

"John is quite shy. His main motivation for studying is to get a better job. He wants to be a bricklayer or joiner. He believes that maths is important for this, but is not so sure about English. One reason for enrolling was seeing a friend who had done the same course get a job. He was disappointed at his performance in the test."

Results

Q. no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
status	r	r	r	w	w	r	w	r	w	r	w	w	r	w	r	w	w	w	w
entered	0	3	2	2	0	2	2	1	2	1	2	0	2	1	2	1	2	0	1

Q. no.	20	21	22	23	24	25	26	27
status	w	u	u	u	u	u	u	u
entered	25							

Report (to be edited)

John Smith,

English Skills

You got eight right. You were best at:

- grammar
- and reading to find things out.

Your friend got a job after doing the course and so can you. If you work hard at English,

- you will find it easier to write job applications
- and you will be more confident in job interviews.

CLICK HERE FOR SOME ENGLISH TIPS

Expert-authored report for John Smith

John got:

- *3 out of 3 Entry 3 questions right*
- *5 out of 12 Level 1 questions right (saw all of them)*
- *0 out of 12 Level 2 questions right (saw 5 of them)*

As he got 5 consecutive questions wrong (16 to 20) he only saw 20 questions in total and this took her 8 minutes, i.e. 24 seconds each question.

One issue for John is that he expressed disappointment at his results. I wonder what he was basing this feeling on. Does he already know what his results mean or did he realise that he was making errors?

Things I might say if he was with me are:

1. *Thanks for doing this.*
2. *Why are you disappointed?*
3. *You answered eight questions correctly.*
4. *What do you feel are your strong points?*
5. *Are there any areas you can think of where you might need some help?*
6. *From this short test I would suggest that you have a few gaps at Level 1. You do seem to be pretty good at finding out what you need to know when you read. Do you agree?*
7. *Would you like to join a class to fill these gaps?*
8. *If you do join a class the teacher will make sure that you can concentrate on what you need to learn. But it would be a good idea to get a little bit more information first on where those gaps are. I would like you to answer some more questions on another day. Are you happy to do that? With the information from that and from what you can tell us we should make sure that you plug those gaps.*
9. *Do you want to ask me anything about the questions or about what happens next?*

Learner Profile #7 (to be written from scratch): Literacy Screener

Name: Diana Hill **Age:** 16 **Her own assessment of her English:** ?

Overall Screener Score: 15 **Time taken:** 16 mins.

Background:

"Diana said that better reading skills would improve her self-confidence and also help her job-wise. She wants to work with children and wants to be able to read stories to them. She did poorly on the reading test, but said she liked to read. She thought she was okay at maths."

Results

Q. No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
status	r	r	w	w	r	r	w	w	r	r	r	r	w	r	r	w	w	r	w
entered	0	3	0	3	2	1	1	2	6	1	0	3	1	3	2	1	1	3	1

Q. No.	20	21	22	23	24	25	26	27
status	w	r	r	w	r	w	r	w
entered	15	1	3	1	0	0	1	3

Expert-authored report for Diana Hill

Diana got:

- *2 out of 3 Entry 3 questions right*
- *8 out of 12 Level 1 questions right (saw all of them)*
- *5 out of 12 Level 2 questions right (saw all of them)*

This took her 20 minutes, i.e. 35 seconds each question.

Things I might say if she was with me (and she was not already attending a class) are:

1. *Thanks for doing this.*
2. *You answered 15 questions correctly.*
3. *You only made mistakes on a couple of questions where you had to read. You said that you like reading – so that does not seem to be a problem for you at all. Do you agree?*
4. *The mistakes you did make were more to do with writing. It may be that you would like to improve your spelling and punctuation.*
5. *What you do next depends on what is important to you. If you do not have any English qualifications you may like to prepare for the national test in English at Level 1.*
6. *Do you want to ask me anything about the questions or about what happens next?*