

Generating student feedback from time-series data using Reinforcement Learning

Dimitra Gkatzia, Helen Hastie, Srinivasan Janarthanam and Oliver Lemon

Department of Mathematical and Computer Sciences

Heriot-Watt University

Edinburgh, Scotland

{dg106, h.hastie, sc445, o.lemon} @hw.ac.uk

Abstract

We describe a statistical Natural Language Generation (NLG) method for summarisation of time-series data in the context of feedback generation for students. In this paper, we initially present a method for collecting time-series data from students (e.g. marks, lectures attended) and use example feedback from lecturers in a data-driven approach to content selection. We show a novel way of constructing a reward function for our Reinforcement Learning agent that is informed by the lecturers' method of providing feedback. We evaluate our system with undergraduate students by comparing it to three baseline systems: a rule-based system, lecturer-constructed summaries and a Brute Force system. Our evaluation shows that the feedback generated by our learning agent is viewed by students to be as good as the feedback from the lecturers. Our findings suggest that the learning agent needs to take into account both the student and lecturers' preferences.

1 Introduction

Data-to-text generation refers to the task of automatically generating text from non-linguistic data (Reiter and Dale, 2000). The goal of this work is to develop a method for summarising time-series data in order to provide continuous feedback to students across the entire semester. As a case study, we took a module in Artificial Intelligence and asked students to fill out a very short diary-type questionnaire on a weekly basis. Questions included, for example, number of deadlines, number of classes attended, severity of personal issues. These data were then combined with the marks from the weekly lab reflecting the students' performance. As data is gathered each week in the

lab, we now have a set of time-series data and our goal is to automatically create feedback. The goal is to present a holistic view through these diary entries of how the student is doing and what factors may be affecting performance.

Feedback is very important in the learning process but very challenging for academic staff to complete in a timely manner given the large number of students and the increasing pressures on academics' time. This is where automatic feedback can play a part, providing a tool for teachers that can give insight into factors that may not be immediately obvious (Porayska-Pomsta and Mellish, 2013). As reflected in NSS surveys¹, students are not completely satisfied with how feedback is currently delivered. The 2012 NSS survey, for all disciplines reported an 83% satisfaction rate with courses, with 70% satisfied with feedback. This has improved from recent years (in 2006 this was 60% for feedback) but shows that there is still room for improvement in how teachers deliver feedback and its content.

In the next section (Section 2) a discussion of the related work is presented. In Section 3, a description of the methodology is given as well as the process of the data collection from students, the template construction and the data collection with lecturers. In Section 4, the Reinforcement Learning implementation is described. In Section 5, the evaluation results are presented, and finally, in Sections 6 and 7, a conclusion and directions for future work are discussed.

2 Related Work

Report generation from time-series data has been researched widely and existing methods have been used in several domains such as weather forecasts (Belz and Kow, 2010; Angeli et al., 2010; Sripada et al., 2004), clinical data summarisation (Hunter

¹<http://www.thestudentsurvey.com/>

et al., 2011; Gatt et al., 2009), narrative to assist children with communication needs (Black et al., 2010) and audiovisual debriefs from sensor data from Autonomous Underwater Vehicles missions (Johnson and Lane, 2011).

The two main challenges for time-series data summarisation are what to say (Content Selection) and how to say it (Surface Realisation). In this work we concentrate on the former. Previous methods for content selection include Gricean Maxims (Sripada et al., 2003); collective content selection (Barzilay and Lapata, 2004); and the Hidden Markov model approach for content selection and ordering (Barzilay and Lee, 2004). NLG systems tend to be very domain-specific and data-driven systems that seek to simultaneously optimize both content selection and surface realisation have the potential to be more domain-independent, automatically optimized and lend themselves to automatic generalization (Angeli et al., 2010; Rieser et al., 2010; Dethlefs and Cuayahuitl, 2011). Recent work on report generation uses statistical techniques from Machine Translation (Belz and Kow, 2010), supervised learning (Angeli et al., 2010) and unsupervised learning (Konstas and Lapata, 2012).

Here we apply Reinforcement Learning methods (see Section 4 for motivation) which have been successfully applied to other NLG tasks, such as Temporal Expressions Generation (Janarthanam et al., 2011), Lexical Choice (Janarthanam and Lemon, 2010), generation of adaptive restaurant summaries in the context of a dialogue system (Rieser et al., 2010) and generating instructions (Dethlefs and Cuayahuitl, 2011).

3 Methodology

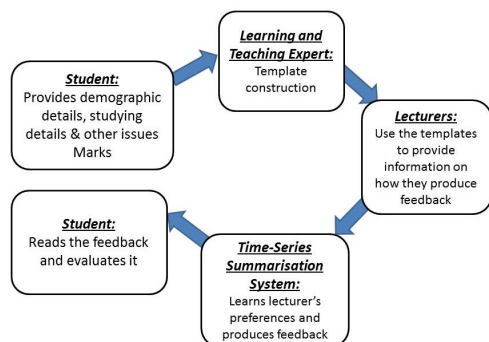


Figure 1: Methodology for data-driven feedback report generation

Figure 1 shows graphically our approach to the development of a generation system. Firstly, we collected data from students including marks, demographic details and weekly study habits. Next, we created templates for surface realisation with the help of a Teaching and Learning expert. These templates were used to generate summaries that were rated by lecturers. We used these ratings to train the learning agent. The output of the learning agent (i.e. automatically generated feedback reports) were finally evaluated by the students. Each of these steps are discussed in turn.

3.1 Time-series Data Collection from Students

The data were collected during the weekly lab sessions of a Computer Science module which was taught to third year Honours and MSc students over the course of a 10 week semester. We recruited 26 students who were asked to fill in a web-based diary-like questionnaire. Initially, we asked students to provide some demographic details (age, nationality, level of study). In addition, students provided on a weekly basis, information for nine factors that could influence their performance. These nine factors were motivated from the literature and are listed here in terms of effort (Ames, 1992), frustration (Craig et al., 2004), difficulty (Person et al., 1995; Fox, 1993) and performance (Chi et al., 2001). *Effort* is measured by three factors: (1) how many hours they studied; (2) the level of revision they have done; (3) as well as the number of lectures (of this module) they attended. *Frustration* is measured by (4) the level of understandability of the content; (5) whether they have had other deadlines; and whether they faced any (6) health and/or (7) personal issues and at what severity. The *difficulty of the lab exercises* is measured by (8) the students' perception of difficulty. Finally, (9) marks achieved by the students in each weekly lab was used as a measure of their *performance*.

3.2 Data Trends

Initially, the data were processed so as to identify the existing trend of each factor during the semester, (e.g. number of lectures attending decreases). The tendencies of the data are estimated using linear least-squares regression, with each factor annotated as INCREASING, DECREASING or STABLE. In addition, for each student we perform a comparison between the average of each

Type	Description	Examples
AVERAGE	describes the factor data by either averaging the values given by the student, or by comparing the student's average with the class average (e.g. if above the mean value for the class, we say that the material is challenging).	"You spent 2 hours studying the lecture material on average". (HOURS STUDIED) "You found the lab exercises very challenging". (DIFFICULTY)
TREND	discusses the trend of the data, e.g. increasing, decreasing or stable.	"Your workload is increasing over the semester". (DEADLINES)
WEEKS	talks about specific events that happened in one or more weeks.	"You have had other deadlines during weeks 5, 6 and 9". (DEADLINES)
OTHER	all other expressions that are not directly related to data.	"Revising material during the semester will improve your performance". (REVISION)

Table 1: The table explains the different template types.

factor and the class average of the same factor.

3.3 Template Generation

The wording and phrasing used in the templates to describe the data were derived from working with and following the advice of a Learning and Teaching (L&T) expert. The expert provided consultation on how to summarise the data. We derived 4 different kinds of templates for each factor: AVERAGE, TREND, WEEKS and OTHER based on time-series data on plotted graphs. A description of the template types is shown in Table 1.

In addition, the L&T expert consulted on how to enhance the templates so that they are appropriate for communicating feedback according to the guidelines of the Higher Education Academy (2009), for instance, by including motivating phrases such as "You may want to plan your study and work ahead".

3.4 Data Collection from Lecturers

The goal of the Reinforcement Learning agent is to learn to generate feedback at least as well as lecturers. In order to achieve this, a second data collection was conducted with 12 lecturers participating.

The data collection consisted of three stages where lecturers were given plotted factor graphs and were asked to:

1. write a free style text summary for 3 students (Figure 2);
2. construct feedback summaries using the templates for 3 students (Figure 3);
3. rate random feedback summaries for 2 students (Figure 4).

We developed the experiment using the Google Web Toolkit for Web Applications, which facil-

itates the development of client-server applications. The server side hosts the designed tasks and stores the results in a datastore. The client side is responsible for displaying the tasks on the user's browser.

In Task 1, the lecturers were presented with the factor graphs of a student (one graph per factor) and were asked to provide a free-text feedback summary for this student. The lecturers were encouraged to pick as many factors as they wanted and to discuss the factors in any order they found useful. Figure 2 shows an example free text summary for a high performing student where the lecturer decided to talk about lab marks and understandability. Each lecturer was asked to repeat this task 3 times for 3 randomly picked students.

In Task 2, the lecturers were again asked to construct a feedback summary but this time they were given a range of sentences generated from the templates (as described in Section 2.3). They were asked to use these to construct a feedback report. The number of alternative utterances generated for each factor varies depending on the factor and the given data. In some cases, a factor can have 2 generated utterances and in other cases up to 5 (with a mean of 3 for each factor) and they differentiate in the style of trend description and wording. Again the lecturer was free to choose which factors to talk about and in which order, as well as to decide on the template style he/she prefers for the realisation through the template options. Figure 3 shows an example of template selection for the same student as in Figure 2.

In Task 3, the lecturers were presented with the plotted factor graphs plus a corresponding feedback summary that was generated by randomly choosing n factors and their templates, and were asked to rate it in a scale between 0-100 (100 for the best summary). Figure 4 shows an example of

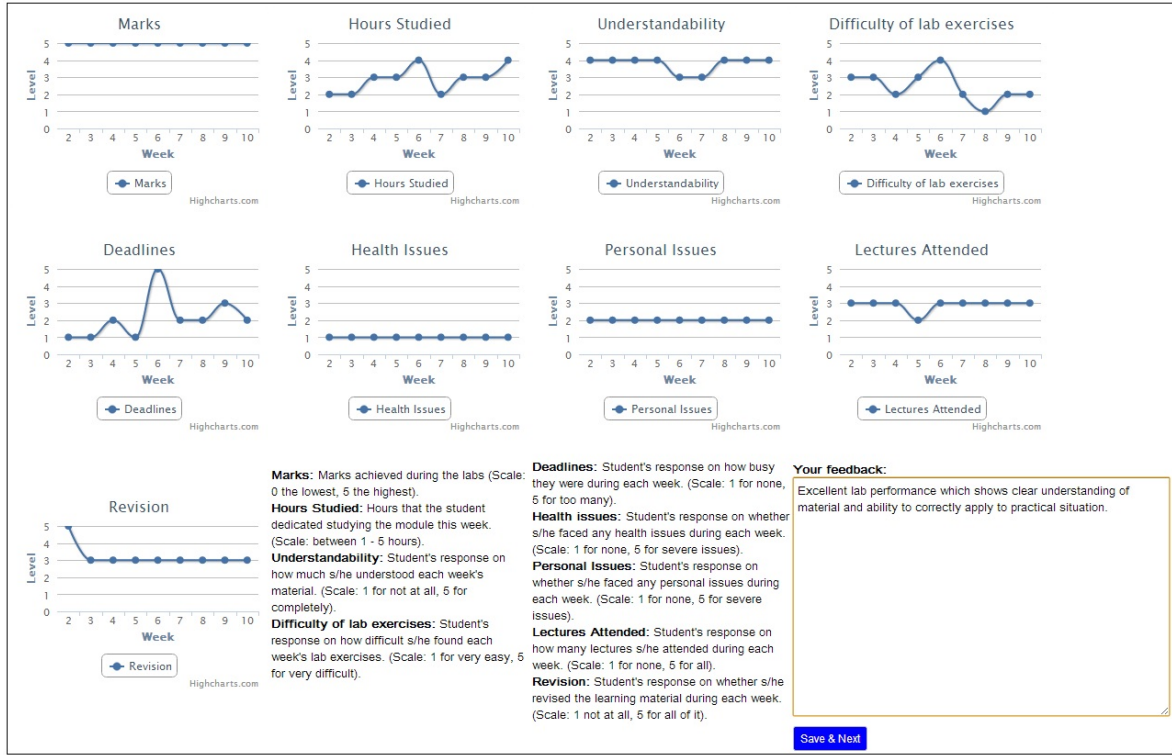


Figure 2: The interface of the 1st task of the data collection: the lecturer consults the factor graphs and provides feedback in a free text format.

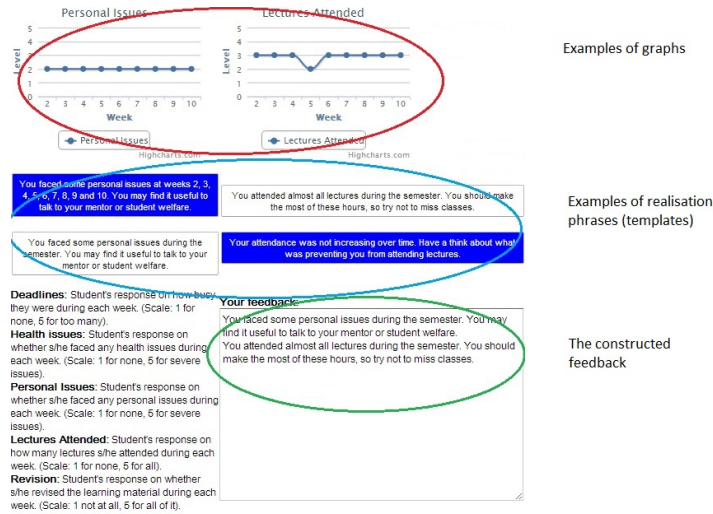


Figure 3: The interface of the 2nd task of data collection: the lecturer consults the graphs and constructs a feedback summary from the given templates (this graph refers to the same student as Figure 2).

a randomly generated summary for the same student as in Figure 2.

4 Learning a Time-Series Generation Policy

Reinforcement Learning (RL) is a machine learning technique that defines how an agent learns to take optimal actions in a dynamic environment so

as to maximize a cumulative reward (Sutton and Barto, 1998). In our framework, the task of content selection of time-series data is presented as a Markov Decision problem. The goal of the agent is to learn to choose a sequence of actions that obtain the maximum expected reward in the long run. In this section, we describe the Reinforcement Learning setup for learning content selection

Here is the automated feedback for this student:

You did not spend much time coping with other deadlines. You could revise the material during the less busy weeks. You did not face any health problems during the semester. Your comprehension of the material is good. Keep up the good work! You faced some personal issues at weeks 2, 3, 4, 5, 6, 7, 8, 9 and 10. You may find it useful to talk to your mentor or student welfare. You dedicated less time studying the lecture material in the beginning of the semester compared to the end of the semester. Keep up the good work!

Rate this summary Save & Next

Figure 4: The interface of the 3rd task of data collection: the lecturer consults the graphs and rates the randomly generated feedback summary (this graph refers to the same student as Figures 2 and 3).

from time-series data for feedback report generation. Summarisation from time-series data is an open challenge and we aim to research other methods in the future, such as supervised learning, evolutionary algorithms etc.

4.1 Actions and States

In this learning setup, we focused only on selecting the correct content, i.e. which factors to talk about. The agent selects a factor and then decides whether to talk about it or not. The state consists of a description of the factor trends and the number of templates that have been selected so far. An example of the initial state of a student can be:

<marks_increased, lectures_attended_stable, hours_studied_increased, understandability_stable, difficulty_increased, health_issues_stable, personal_issues_stable, revision_increased, 0>

The agent explores the state space by selecting a factor and then by deciding whether to talk about it or not. If the agent decides to talk about the selected factor, it chooses the template in a greedy way, i.e. it chooses for each factor the template that results in a higher reward. After an action has been selected, it is deleted from the action space.

4.1.1 Ordering

In order to find out in which order the lecturers describe the factors, we transformed the feedback summaries into n-grams of factors. For instance, a summary that talks about the student's performance, the number of lectures that he/she attended, potential health problems and revision

done can be translated into the following ngram: start, marks, lectures_attended, health_issues, revision, end. We used the constructed n-grams to compute the bigram frequency of the tokens in order to identify which factor is most probable to be referred to initially, which factors follow particular factors and which factor is usually talked about in the end. It was found that the most frequent ordering is: start, marks, hours_studied, understandability, difficulty, deadlines, health_issues, personal_issues, lectures_attended, revision, end.

4.2 Reward Function

The goal of the reward function is to optimise the way lecturers generate and rate feedback. Given the expert annotated summaries from Task 1, the constructed summaries from Task 2 and the ratings from Task 3, we derived the multivariate reward function:

$$Reward = a + \sum_{i=1}^n b_i * x_i + c * length$$

where $X = \{x_1, x_2, \dots, x_n\}$ represents the combinations between the data trends observed in the time-series data and the corresponding lecturers' feedback (i.e. whether they included a factor to be realised or not and how). The value x_i for factor i is defined by the function:

$$x_i = \begin{cases} 1, & \text{the combination } i \text{ of a factor trend} \\ & \text{and a template type is included in} \\ & \text{the feedback} \\ 0, & \text{if not.} \end{cases}$$

For instance, the value of x_1 is 1 if marks were increased and this trend is realised in the feedback, otherwise it is 0. In our domain $n = 90$ in order to cover all the different combinations. The *length* stands for the number of factors selected, a is the intercept, b_i and c are the coefficients for x_i and *length* respectively.

In order to model the reward function, we used linear regression to compute the weights from the data gathered from the lecturers. Therefore, the reward function is fully informed by the data provided by the experts. Indeed, the intercept a , the vector weights b and the weight c are learnt by making use of the data collected by the lecturers from the 3 tasks discussed in Section 3.4.

The reward function is maximized (Reward = 861.85) for the scenario (i.e. each student's data), content selection and preferred template style shown in Table 2 (please note that this scenario was not observed in the data collection).

Factor	Trend	Template
difficulty	stable	NOT_MENTIONED
hours studied	stable	TREND
understandability	stable	NOT_MENTIONED
deadlines	increase	WEEKS
health issues	stable	WEEKS
personal issues	stable	WEEKS
lectures att.	stable	WEEKS
revision	stable	OTHER
marks	increase	TREND

Table 2: The table shows the scenario at which the reward function is maximised.

The reward function is minimized (Reward = -586.0359) for the scenario shown in Table 3 (please note that this scenario also was not observed in the data collection).

Factor	Trend	Template
difficulty	increase	AVERAGE
hours studied	stable	NOT_MENTIONED
understandability	decrease	AVERAGE
deadlines	*	*
health issues	increase	TREND
personal issues	stable	TREND
lectures att.	stable	NOT_MENTIONED
revision	stable	AVERAGE
marks	stable	TREND

Table 3: The table shows the scenario at which the reward function is minimised (* denotes multiple options result in the same minimum reward).

4.3 Training

We trained a time-series generation policy for 10,000 runs using the Tabular Temporal-Difference Learning (Sutton and Barto, 1998). During the training phase, the learning agent generated feedback summaries. When the construction of the summary begins, the length of the summary is 0. Each time that the agent adds a template (by selecting a factor), the length is incremented, thus changing the state. It repeats the process until it decides for all factors whether to talk about them or not. The agent is finally rewarded at the end of the process using the Reward function described in Section 3.2. Initially, the learning agent selects factors randomly, but gradually learns to identify factors that are highly rewarding for a given data scenario. Figure 5 shows the learning curve of the agent.

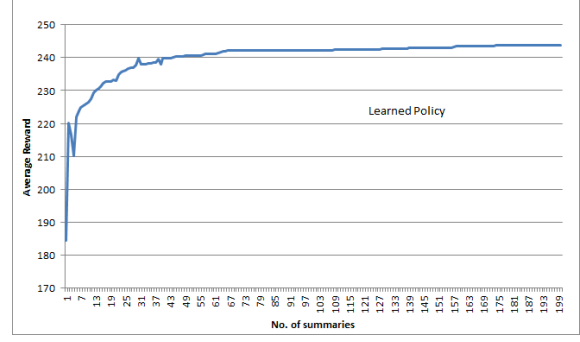


Figure 5: Learning curve for the learning agent. The x-axis shows the number of summaries produced and y- axis the total reward received for each summary.

5 Evaluation

We evaluated the system using the reward function and with students. In both these evaluations, we compared feedback reports generated using our Reinforcement Learning agent with four other baseline systems. Here we present a brief description of the baseline systems.

Baseline 1: Rule-based system. This system selects factors and templates for generation using a set of rules. These hand-crafted rules were derived from a combination of the L&T expert’s advice and a student’s preferences and is therefore a challenging baseline and represents a middle ground between the L&T expert’s advice and a student’s preferences. An example rule is: if the mark average is less than 50% then refer to revision.

Baseline 2: Brute Force system. This system performs a search of the state space, by exploring randomly as many different feedback summaries as possible. The Brute Force algorithm is shown below:

Algorithm 1 Brute Force algorithm

```

Input data: D
for n=0...10,000
  construct randomly feedback[n]
  assign getReward[n]
  if getReward[n]>getReward[n-1]
    bestFeedback = feedback[n]
  else
    bestFeedback = feedback[n-1]
return bestFeedback

```

In each run the algorithm constructs a feedback summary, then it calculates its reward, using the same reward function used for the Reinforcement Learning approach, and if the reward of the new feedback is better than the previous, it keeps the

new one as the best. It repeats this process for 10,000 times for each scenario. Finally, the algorithm returns the summary that scored the highest ranking.

Baseline 3: Lecturer-produced summaries.

These are the summaries produced by the lecturers, as described in Section 2.4, for Task 2 using template-generated utterances.

Baseline 4: Random system: The Random system constructs feedback summaries by selecting factors and templates randomly as described in Task 3 (in Section 3.4).

5.1 Evaluation with Reward Function

Table 4 presents the results of the evaluation performed using the Reward Function, comparing the learned policy with the four baseline systems. Each system generated 26 feedback summaries. On average the learned policy scores significantly higher than any other baseline for the given scenarios ($p < 0.05$ in a paired t-test).

Time-Series Summarisation Systems	Reward
Learned	243.82
Baseline 1: Rule-based	107.77
Baseline 2: Brute Force	241.98
Baseline 3: Lecturers	124.62
Baseline 4: Random	43.29

Table 4: The table summarises the average rewards that are assigned to summaries produced from the different systems.

5.2 Evaluation with Students

A subjective evaluation was conducted using 1st year students of Computer Science as participants. We recruited 17 students, who were all English native speakers. **The participants were shown 4 feedback summaries in a random order**, one generated by the learned policy, one from the rule-based system (Baseline 1), one from the Brute Force system (Baseline 2) and one summary produced by a lecturer using the templates (Baseline 3). Given the poor performance of the Random system in terms of reward, Baseline 4 was omitted from this study.

Overall there were 26 different scenarios, as described in Section 3.1. All summaries presented to a participant were generated from the same scenario. **The participants then had to rank the summaries in order of preference: 1 for the most preferred and 4 for the least preferred.** Each partici-

pant repeated the process for 4.5 scenarios on average (the participant was allowed to opt out at any stage). The mode values of the rankings of the preferences of the students are shown in Table 5. The web-based system used for the evaluation is shown in Figure 6.

System	Mode of Rankings
Learned	3rd
Baseline 3: Lecturers	3rd
Baseline 1: Rule-based	1st
Baseline 2: Brute Force	4th

Table 5: The table shows the mode value of the rankings of the preference of the students.

We ran a **Mann-Whitney’s U test** to evaluate the difference in the responses of our 4-point Likert Scale question between the Learned system and the other three baselines. It was found that, for the given data, the preference of students for the feedback generated by the Learned system is as good as the feedback produced by the experts, i.e. there is no significant difference between the mean value of the rankings of the Learned system and the lecturer-produced summaries ($p = 0.8$) (Baseline 3).

The preference of the users for the Brute Force system does not differ significantly from the summaries generated by the Learned system ($p = 0.1335$). However, the computational cost of the Brute Force is higher because each time that the algorithm sees a new scenario it has to run approximately 3k times to reach a good summary (as seen in Figure 7) and about 10k to reach an optimal one, which corresponds to 46 seconds. This delay would prohibit the use of such a system in time-critical situations (such as defence) and in live systems such as tutoring systems. In addition, the processing time would increase with more complicated scenarios and if we want to take into account the ordering of the content selection and/or if we have more variables. In contrast, the RL method needs only to be trained once.

Finally, the users significantly preferred the summaries produced by the Rule-based system (Baseline 1) to the summaries produced by the Learned system. This is maybe because of the fact that in the rule-based system some knowledge of the end user’s preferences (i.e. students) was taken into account in the rules which was not the case in the other three systems. This fact suggests that

Imagine you are this student, please rank these paragraphs in order of preference: 1st being the most effective format for feedback, 4th being the least effective format for feedback.

<p>You did well at weeks 2, 3 and 5, but not at weeks 4, 6, 7, 8, 9 and 10. Have a think about how you were working well and try to apply it to the other labs. You spent 1.2 hours studying the lecture material on average. You should dedicate more time to study. Your understanding of the material could be improved. Try going over the teaching material again. Your attendance was not increasing over time. Have a think about what was preventing you from attending lectures. Your workload is increasing over the semester. You may want to plan your studying and work in advance. Revising material during the semester will improve your performance in the lab.</p>	<p>You did well at weeks 2, 3 and 5, but not at weeks 4, 6, 7, 8, 9 and 10. Have a think about how you were working well and try to apply it to the other labs. You attended all lectures during the semester. Have a think about how to use time in lectures to improve your understanding of the material. You found the level of difficulty of the lab exercises of average difficulty. You could try out some more advanced material and exercises. You dedicated more time studying the lecture material in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. You should study harder to improve your comprehension of the material. Try going over the teaching material again. Your workload is increasing over the semester. You may want to plan your studying and work in advance. You revised part of the learning material. Have a think whether revising has improved your performance.</p>	<p>Your overall performance has improved since the beginning of the semester. Keep up the good work and maybe try some more challenging exercises. You dedicated more time studying the lecture material in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. You found the lab exercises not very challenging. You could try out some more advanced material and exercises. Your workload is increasing over the semester. You may want to plan your studying and work in advance. You revised part of the learning material. Have a think whether revising has improved your performance. Your health condition remained stable during the semester.</p>	<p>Your overall performance has improved since the beginning of the semester. Keep up the good work and maybe try some more challenging exercises. You dedicated more time studying the lecture material in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. You found the lab exercises not very challenging. You could try out some more advanced material and exercises. Your workload is increasing over the semester. You may want to plan your studying and work in advance. You revised part of the learning material. Have a think whether revising has improved your performance. Your health condition remained stable during the semester.</p>
rank this summary <input type="text" value="1"/>	rank this summary <input type="text" value="1"/>	rank this summary <input type="text" value="1"/>	rank this summary <input type="text" value="1"/>
<input type="button" value="Submit Ratings"/>			

Figure 6: The interface for the evaluation: the students viewed the four feedback summaries and ranked them in order of preference. From left to right, the summaries as generated by: an Expert (Baseline 3), the Rule based system (Baseline 1), the Brute Force algorithm (Baseline 2), the Learned system.

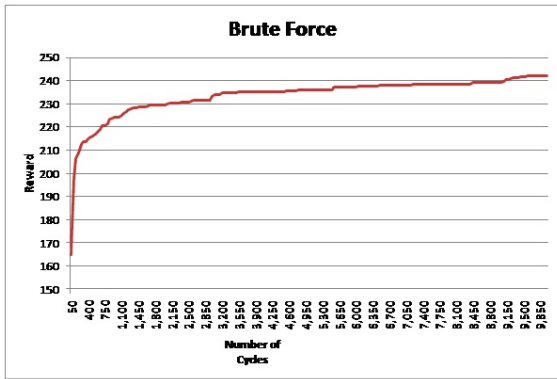


Figure 7: The graphs shows the number of cycles that the Brute Force algorithm needs to achieve specific rewards.

students' preferences should be taken into account as they are the receivers of the feedback. This can also be generalised to other areas, where the experts and the end users are not the same group of people. As the learned policy was not trained to optimise for the evaluation criteria, in future, we will explore reward functions that bear in mind both the expert knowledge and the student's preferences.

6 Conclusion

We have presented a statistical learning approach to summarisation from time-series data in the area of feedback reports. In our reports, we took into

account the principles of good feedback provision as instructed by the Higher Education Academy. We also presented a method for data gathering from students and lecturers and show how we can use these data to generate feedback by presenting the problem as a Markov Decision Process and optimising it using Reinforcement Learning techniques. We also showed a way of constructing a data-driven reward function that can capture dependencies between the time-series data and the realisation phrases, in a similar way that the lecturers do when providing feedback. Finally, our evaluation showed that the learned report generation policy generates reports as well as lecturers.

7 Future Work

We aim to conduct further qualitative research in order to explore what factors and templates students find useful to be included in the feedback and inform our reward function with this information as well as what we have observed in the lecturer data collection. This way, we hope, not only to gain insights into what is important to students and lecturers but also to develop a data-driven approach that, unlike the rule-based system, does not require expensive and difficult-to-obtain expert input from Learning and Teaching experts. In addition, we want to compare RL techniques with supervised learning approaches and evolutionary algorithms. Finally, we want to unify content se-

lection and surface realisation, therefore we will extend the action space in order to include actions for template selection.

8 Acknowledgements

The research leading to this work has received funding from the EC's FP7 programme: (FP7/2011-14) under grant agreement no. 248765 (Help4Mood).

References

- Carole Ames. 1992. Classrooms: Goals, Structures, and Student Motivation. *Journal of Educational Psychology*, 84(3):p261-71.
- Gabor Angeli, Percy Liang and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. *EMNLP '10: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.
- Regina Barzilay and Mirella Lapata. 2004. Collective content selection for concept-to-text generation. *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*.
- Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. *HLT-NAACL 2004: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.
- Anja Belz and Eric Kow. 2010. Extracting parallel fragments from comparable corpora for data-to-text generation. *INLG '10: Proceedings of the 6th International Natural Language Generation Conference*.
- Rolf Black, Joe Reddington, Ehud Reiter, Nava Tintarev, and Annalu Waller. 2010. Using NLG and Sensors to Support Personal Narrative for Children with Complex Communication Needs. *SLPAT '10: Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*.
- Micheline T.H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, Robert G. Hausmann. 2001. Learning from human tutoring. *Journal of Cognitive Science*, 25(4):471-533.
- Scotty D. Craig, Arthur C. Graesser, Jeremiah Sullins, Barry Gholson. 2004. Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, 29:241-250.
- Nina Dethlefs and Heriberto Cuayahuitl. 2011. Combining hierarchical reinforcement learning and bayesian networks for natural language generation in situated dialogue. *ENLG '11: Proceedings of the 13th European Workshop on Natural Language Generation*.
- Barbara Fox. 1993. *The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Albert Gatt, Francois Portet, Ehud Reiter, James Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From Data to Text in the Neonatal Intensive Care Unit: Using NLG Technology for Decision Support and Information Management. *Journal of AI Communications*, 22:153-186.
- Higher Education Academy. 2009. Providing individual written feedback on formative and summative assessments. http://www.heacademy.ac.uk/assets/documents/resources/database/id353_senleef_guide.pdf. Last modified September 16.
- Jim Hunter, Yvonne Freer, Albert Gatt, Yaji Sripada, Cindy Sykes, and D Westwater. 2011. BT-Nurse: Computer Generation of Natural Language Shift Summaries from Complex Heterogeneous Medical Data. *Journal of the American Medical Informatics Association*, 18:621-624.
- Srinivasan Janarthenam, Helen Hastie, Oliver Lemon, Xingkun Liu. 2011. "The day after the day after tomorrow?" A machine learning approach to adaptive temporal expression generation: training and evaluation with real users. *SIGDIAL '11: Proceedings of the SIGDIAL 2011 Conference*.
- Srinivasan Janarthanam and Oliver Lemon. 2010. Adaptive Referring Expression Generation in Spoken Dialogue Systems: Evaluation with Real Users. *SIGDIAL '10: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.
- Nicholas A. R. Johnson and David M. Lane. 2011. Narrative Monologue as a First Step Towards Advanced Mission Debrief for AUV Operator Situational Awareness. In the 15th International Conference on Advanced Robotics.
- Ioannis Konstas and Mirella Lapata. 2012. Unsupervised concept-to-text generation with hypergraphs. *NAACL HLT '12: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Natalie K. Person, Roger J. Kreuz, Rolf A. Zwaan and Arthur C. Graesser. 1995. Pragmatics and Pedagogy: Conversational Rules and Politeness Strategies May Inhibit Effective Tutoring. *Journal of Cognition and Instruction*, 13(2):161-188.
- Kaska Porayska-Pomsta and Chris Mellish. 2013. Modelling human tutors' feedback to inform natural language interfaces for learning. *International Journal of Human-Computer Studies*, 71(6):703724.

- Ehud Reiter and Robert Dale. 2000. Building Natural Language Generation systems. Cambridge University Press.
- Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising Information Presentation for Spoken Dialogue Systems. ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.
- Somayajulu Sripada, Ehud Reiter, I Davy, and K Nilssen. 2004. Lessons from Deploying NLG Technology for Marine Weather Forecast Text Generation. In Proceedings of PAIS session of ECAI-2004:760-764.
- Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2003. Generating English Summaries of Time Series Data using the Gricean Maxims. KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.
- Richart Sutton and Andrew Barto. 1998. Reinforcement Learning. MIT Press.