Abstract

We present FeedbackGen, a system that uses a multi-adaptive approach to Natural Language Generation. With the term ‘multi-adaptive’, we refer to a system that is able to adapt its content to different user groups simultaneously, in our case adapting to both lecturers and students. We present a novel approach to student feedback generation, which simultaneously takes into account the preferences of lecturers and students when determining the content to be conveyed in a feedback summary. In this framework, we utilise knowledge derived from ratings on feedback summaries by extracting the most relevant features using Principal Component Regression (PCR) analysis. We then model a reward function that is used for training a Reinforcement Learning agent. Our results with students suggest that, from the students’ perspective, such an approach can generate more preferable summaries than a purely lecturer-adapted approach.

1 Introduction

Summarisation of time-series data refers to the task of automatically generating reports from attributes whose values change over time. Content selection is the task of choosing what to say, i.e. what information is to be included in a report (Reiter and Dale, 2000). We consider the task of automatically generating feedback summaries for students describing their performance during the lab of a computer science module over the semester.

Various factors can influence students’ learning such as difficulty of the material (Person et al., 1995), workload (Craig et al., 2004), attendance in lectures (Ames, 1992), etc. These factors change over time and can be interdependent.

In addition, different stakeholders often have conflicting goals, needs and preferences, for example managers with employees, or doctors with patients and relatives, or novice and expert users. In our data, for instance, lecturers tend to comment on the hours that the student studied, whereas the students disprefer this content. In our previous work, we showed that lecturers and students have different perceptions regarding what constitutes good feedback (Gkatzia et al., 2013). Here, we present a novel approach to generation by adapting its content to two user groups simultaneously. Producing the same summary for two groups is important as it allows for shared context and meaningful further discussion and reduces development time.

2 Related Work

Previous work on NLG systems that address more than one user group employs different versions of a system for each different user group (Gatt et al., 2009; Hunter et al., 2011; Mahamood and Reiter, 2011), makes use of User Models (Janarthanam and Lemon, 2010; Thompson et al., 2004; Zukerman and Litman, 2001) or personalises the output to individual users using rules (Reiter et al., 1999). Our proposed system adapts the output to the preferences of more than one user type\(^1\), lecturers and students, but instead of developing many different systems or using User Models that describe different users, it attempts to model the middle ground between the preferences.

In order to identify the users’ preferences, we apply Principal Components Regression (PCR (Jolliffe, 1982)) analysis to two datasets that contain lecturers’ and students’ ratings and identify the most important variables from the principal components, which are then included in a reward function. This hand-crafted reward function is used for training an RL agent for summarisation.

\(^1\)Our approach is different to multi-objective optimisation.
of time-series data. Our previous work showed that when comparing RL and supervised learning in the context of student feedback generation, students preferred the output generated by the RL system (Gkatzia et al., 2014a). Therefore, here, we used RL rather than a supervised learning method. The work described here builds on work reported in (Gkatzia et al., 2014b), which uses as a reward function the average of the Lecturer-adapted and Student-adapted reward functions. However, that method seems to cancel out the preferences of the two groups whereas PCR is able to identify relevant content for both groups.

In the next section, we describe the data used, and the methodology for the multi-adaptive NLG, as well as two alternative systems. In Section 4, we describe the comparison of these three systems in a subjective evaluation and present the results in Section 5. A discussion follows in Section 6 and finally, future work is discussed in Section 7.

3 Methodology

Reinforcement Learning is a machine learning technique that defines how an agent learns to take optimal sequences of actions so as to maximize a cumulative reward (Sutton and Barto, 1998). In our framework, the task of summarisation of time-series data is modelled as a Markov Decision Process, where the decisions on content selection correspond to a sequence of actions (see Section 3.2). Temporal Difference (TD) learning (Sutton and Barto, 1990) is used for training three agents in a simulated environment to learn to make optimal content selection decisions:

1. by adapting to both groups simultaneously,
2. by adapting to lecturers,
3. by adapting to students.

3.1 The Data

For this study, the dataset described in (Gkatzia et al., 2013) was used. Table 1 presents an example of this dataset that describes a student’s learning factors and an aligned feedback summary provided by a lecturer. The dataset is composed of 37 similar instances. Each instance consists of time-series information about the student’s learning routine and the selected templates that lecturers used to provide feedback to this particular student. A template is a quadruple consisting of an id, a factor (bottom left of Table 1), a reference type (trend, week, average, other) and surface text. For instance, a template can be (1, marks, trend, “Your marks were <trend> over the semester”). The lexical choice for <trend> (i.e. increasing or decreasing) depends on the values of time-series data. There is a direct mapping between the values of factor
and reference type and the surface text. The time-series factors are listed in Table 1.

3.2 Actions and states
The state consists of the time-series data and the number of factors which have so far been selected to be talked about (the change of the value of this variable consequently introduces a state change). In order to explore the state space the agent selects a time-series factor (e.g. marks, deadlines etc.) and then decides whether to talk about it or not, until all factors have been considered.

3.3 Reward function
The reward function is the following cumulative multivariate function:

\[ \text{Reward} = a + \sum_{i=1}^{n} b_i \cdot x_i + c \cdot \text{length} \]

where \( X = \{x_1, x_2, ..., x_n\} \) describes the chosen combinations of the factor trends observed in the time-series data and a particular template (i.e. the way of mentioning a factor). \( a, b \) and \( c \) are the correlation coefficients and \( \text{length} \) describes the number of factors selected to be conveyed in the feedback summary. The value of \( x_i \) is given by the function:

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

The coefficients represent the level of preference for a factor to be selected and the way it is conveyed in the summary. In the training phase, the agent selects a factor and then decides whether to talk about it or not. If the agent decides to refer to a factor, the selection of the template is then performed in a deterministic way, i.e. it selects the template that results in higher reward.

Each rated summary is transformed into a vector of 91 binary features. Each feature describes both (1) the trend of a factor (e.g. marks increasing, see also Table 1) and (2) the way that this factor could be conveyed in the summary (e.g. one possible way is referring to average, another possible way is referring to increasing/decreasing trend). If both conditions are met, the value of the feature is 1, otherwise 0. The 91 binary features describe all the different possible combinations. For both the Lecturer-adapted and Student-adapted systems, the reward function is derived from a linear regression analysis of the provided dataset, similarly to Walker et al. (1997) and Rieser et al. (2010).

3.3.1 Multi-adaptive Reward Function
In order to derive a reward function that finds a balance between the two above mentioned systems, we use PCR to reduce the dimensionality of the data and thus reduce the introduced noise. Through PCR we are able to reduce the number of features and identify components of factors that are deemed important to both parties to be used in the reward function.

PCR is a method that combines Principal Component Analysis (PCA) (Jolliffe, 1986) with linear regression. PCA is a technique for reducing the dataset dimensionality while keeping as much of the variance as possible. In PCR, PCA is initially performed to identify the principal components, in our case, the factors that contribute the most to the variance. Then, regression is applied to these principal components to obtain a vector of estimated coefficients. Finally, this vector is transformed back into the general linear regression equation. After performing this analysis on both datasets (students and lecturers), we choose the most important (i.e. the ones that contribute the most to the variance) common components or features resulting in 18 features which were used in the reward function. We then design a handcrafted reward function taking into account this PCR analysis. The five most important features are shown in Table 2.

<table>
<thead>
<tr>
<th>factor trend</th>
<th>way it is mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) marks stable</td>
<td>average</td>
</tr>
<tr>
<td>(2) hours studied decreasing</td>
<td>trend</td>
</tr>
<tr>
<td>(3) health_issues decreasing</td>
<td>weeks</td>
</tr>
<tr>
<td>(4) lectures_attended stable</td>
<td>average</td>
</tr>
<tr>
<td>(5) personal_issues increasing</td>
<td>trend</td>
</tr>
</tbody>
</table>

Table 2: The top 5 features out of the 18 selected through PCR analysis.

4 Evaluation
FeedbackGen is evaluated with real users against two alternative systems: one that adapts to lecturers’ preferences and one that adapts to students’ preferences. The output of the three systems is ranked by 30 computer science students from a variety of years of study. Time-series data of three students are presented on graphs to each participant, along with three feedback summaries (each one generated by a different system), in random order, and they are asked to rank them in terms of preference.
Table 3: The table presents example outputs from the three different systems in order of highest ranked (bold signifies the chosen template content, * denotes significance with p <0.05 after comparing each system with each other using Mann Whitney U test).

Table 3: The table presents example outputs from the three different systems in order of highest ranked (bold signifies the chosen template content, * denotes significance with p <0.05 after comparing each system with each other using Mann Whitney U test).

5 Results
Table 3 shows three summaries that have been generated by the different systems. As we can see from Table 3, students significantly prefer the output of the system that is trained for their preferences. In contrast, students significantly disprefer the system that is trained for lecturers’ preferences. Finally, they rank as second the system that captures the preferences of both lecturers and students, which shows that it might be feasible to find middle ground between the preferences of two user groups. Significance testing is done using a Mann Whitney U test (p <0.05), performing a pair-wise comparison.

6 Discussion
The weights derived from the linear regression analysis vary from the Lecturer-adapted function to the Student-adapted function. For instance, the lecturers’ most preferred content is hours studied. This, however, does not factor heavily into the student’s reward function, apart from the case where hours studied are decreasing or remain stable (see also Table 2).

Students like reading about personal issues when the number of issues they faced was increasing over the semester. On the other hand, lecturers find it useful to give advice to all students who faced personal issues during the semester, hence personal issues are included in the top 18 features (Table 2). Moreover, students seem to mostly prefer a feedback summary that mentions the understandability of the material when it increases, which is positive feedback.

As reflected in Table 2, the analysis of PCR showed that both groups found it useful to refer to the average of marks when they remain stable. In addition, both groups found understandability when it increases useful, for a variety of reasons, for example lecturers might find it useful to encourage students whereas students might prefer to receive positive feedback. Both groups also agree on hours studied as described earlier. On the other hand, both groups find mentioning the students’ difficulty when it decreases as positive.

7 Future Work
In the future, we plan to evaluate our methodology with lecturers and a larger sample of students across different disciplines. Moreover, we aim to port our methodology to a different domain, and try to find the middle ground between the preferences of novices and expert users when summarising medical data while providing first aid. Finally, we want to compare the methodology presented here to a multi-objective optimisation approach (Fonseca and Flemming, 1993), where the preferences of each user group will be modelled as two different optimisation functions.

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


Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. 2014b. Finding Middle Ground? Multi-objective Natural Language Generation from Time-series data. In *14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.


