Toward Empirical Analysis of Pedagogical Feedback in Computer Programming Learning Environments

Gabriel Raubenheimer  
School of Electrical and Information Engineering  
The University of Sydney  
Sydney, NSW, Australia  
grau0521@uni.sydney.edu.au

Bryn Jeffries  
Grok Learning  
Sydney, NSW, Australia  
bryn@groklearning.com

Kalina Yacef  
School of Computer Science  
The University of Sydney  
Sydney, NSW, Australia  
kalina.yacef@sydney.edu.au

ABSTRACT
Digital learning environments are emerging as a key part of the future of computer science education. However, there is little empirical understanding of what forms of didactic feedback are pedagogically optimal for short- and long-term learning outcomes in these new contexts. Methods for classification of feedback in this new context are thus needed, to enable empirical analysis of what constitutes effectiveness. Whilst numerous taxonomies of feedback exist, they do not provide suitable classification for assessing impact of feedback approaches on student learning. We provide an empirically and theoretically meaningful framework for analysing feedback in digital learning environments. The classification is based on placement along two axes – whether feedback is problem or solution centric, and whether it provides information pertaining to a specific instance of a student’s work or generalised to the underlying theory. We apply this framework to analyse feedback given in an online computer programming course, showing that types of feedback provided effect attainment of short-term goal-oriented student outcomes. This motivates its possible application in understanding more long-term acquisition and retention of knowledge, both in computer science education and beyond.

CCS CONCEPTS
• Applied Computing – Education - Computer-assisted Instruction  
• Social Professional Topics - Computing Education

KEYWORDS
Feedback taxonomy, feedback analysis, theory of learning, instructivism, constructivism, digital learning environments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

1 Introduction
Feedback is critical to students’ learning outcomes [1]. As educators increasingly engage with students through digital learning environments, it is important to understand how feedback can effectively be provided in these rapidly developing contexts. These environments have also made that data more available. Feedback is an important means for influencing students’ attainment, motivation and attitudes and research into feedback in learning science has repeatedly illustrated its importance. However, despite the presence of a number of well-founded taxonomies and frameworks to understand the subject, the literature remains inconclusive on what types of feedback best helps students achieve learning outcomes. To address this gap, we propose a taxonomy of feedback for digital environments, through which we aim to empirically analyse the effectiveness of different forms of feedback to understand how to best support students’ short- and long-term learning outcomes. Closing this gap is particularly prescient during this time of proliferating digital learning environments, as current taxonomies do not allow us to compare the effectiveness of feedback with different pedagogical underpinnings. Without the comparison this taxonomy aims to facilitate, we can’t empirically select pedagogical paradigms upon which to build these new environments.

The secondary motivation for creation of a new taxonomy of feedback is to address the apparent gap between the literatures of learning science and learning analytics. Techniques and research in educational data mining have produced deep and novel insight into the experience and attainment of learners. These results, however, are seldom tied to pedagogical theories [2], reducing the opportunity for them to contribute to the learning science literature. Particularly with the current proliferation of digital learning environments, we sought to develop a theory-driven method for classification of feedback to aid the analysis of educational outcomes both in these new contexts and more generally within educational theory.

Computer science education has a particular opportunity in this regard. As well as being on the vanguard of education digitisation,
digital learning environments in this field generate the readily interpretable data needed for empirical analysis. Furthermore, educational data mining has an established history of insight in this area, bringing with it a rich set of analytical methods. Finally, whilst the default mode of educational delivery of mathematics, languages and science has long been face-to-face, computer science education in schools is more likely to be digitally facilitated.

The focus of this paper is a proposed framework used to categorise and thus analyse feedback in digital computer programming learning environments. Whilst many taxonomies of feedback provide either extremely fine-grained or contextually inappropriate classification methods, we classify along two simple and readily observable axes: whether the feedback is problem or solution oriented, and whether it pertains to a specific instance of a student’s mistake or to the underlying theory or concepts that underly the error. We then use this framework to categorise feedback provided by human tutors to students in a digital programming learning program, Grok Learning1, and measure the effectiveness of this feedback. In this analysis, we use the first week of the NCSS Intermediate Challenge 2019 aligned to the Year 9/10 Australian federal Digital Technologies curriculum, and analyse feedback effectiveness through two indicators: the amount of submissions taken for students to make progress after receiving feedback, and the amount of feedback required in a given intervention.

2 Related Work

There is strong evidence for feedback having a significant positive effect on learning attainment [3], although certain forms of feedback have been shown to be consistently detrimental [4]. Attempts to distinguish between effective and ineffective feedback have been done through lenses such as the different cognitive mechanisms employed, varying levels of specificity and the varying purposes feedback is designed for [1]. Current taxonomies include an attempt to understand students’ reactions to teachers [5], and one to create automated hints for learners [6]. Other analysis has been done through more general frameworks for understanding feedback. For example, Black and Wiliam [3] specified that feedback must address where the learner is right now, where the learner is going, and how they might get there. Similarly, Hattie and Timperley [7] delineated feedback into the four levels of ‘feedback about the task’, ‘feedback about the processing of the task’, ‘feedback about self-regulation’ and ‘feedback about the self as a person’. In computer science education, modelling of student progression through tasks has been used to inform feedback generated by automated tutors [8, 9], perform problem generation [10], and to help educators understand when teacher intervention might be required [11]. This work provides a solid theoretical foundation on the role, importance and potential for classification of feedback, as well as the application of empirical methods to understand and assist learners in digital environments. Upon this foundation, methods can be created to understand empirically and theoretically the role and effectiveness of feedback in digital environments, and to help teachers choose appropriate methods for provision of feedback therein.

3 Quadrants for Analysis of Feedback

We aim to be able to categorise feedback from a range of digital computer science learning environments, such that we might empirically evaluate the effect of this feedback. We need to identify features that are readily recognisable in feedback from these environments and allow for analysis of this feedback with respect to seminal educational theories and frameworks. In doing so, we hope to demonstrate that bridging the fields of learning analytics and learning science presents an opportunity to enrich both fields.

A theoretical framework that is of particular interest for understanding feedback in computer science education is the distinction between the constructivist and instructivist learning paradigms. The impact of constructivism has proven difficult to directly measure, and there is little empirical evidence to show it leads to consistently better learning outcomes compared to its counterpart. According to Richardson [12], one reason for this is that the comparison of results between the two lacks meaning because, in traditional learning environments, the goals of constructivist methodologies are different to those of its instructivist methodologies. We would question, however, whether digital environments share these goals, particularly in computer programming education, which is heavily oriented toward goal-oriented achievement and skills acquisition. As such, we suggest that the comparison between these methodologies is newly meaningful in the given context. Meanwhile, instructivism has been shown to limit exploratory engagement [13], but also to increase the number of students who achieve understanding of concepts [14]. This is pertinent to us – one of the promises of digital learning environments (digital anything) is scalability, inherent within which is the minimisation of teaching resources relative to number of students. When a pedagogical system can be constructed that works for most students, it minimises extra intervention required to make it work for all students. Applying this to provision of feedback, the question arises of delineation between ‘constructivist feedback’ and ‘instructivist feedback’.

This delineation is multivariate and nuanced, as digital learning environments in computer science education are unlikely to be categorically either constructivist or instructivist. As such we specify what we believe to be constructivist and instructivist approaches to feedback therein. According to Elliot et al., constructivism is “an approach to learning that holds that people actively construct or make their own knowledge and that reality is determined by the experiences of the learner” [15]. Accordingly,

1 https://groklearning.com/
we characterise a constructivist approach to providing feedback as one where the tutor becomes a part of the environment that the learner is experiencing and constructing within – they form part of the landscape of interactions through which the learner constructs their understanding (e.g., ‘variable assignment on line 4 is being overridden in line 6’). They facilitate students’ own construction of knowledge rather than directly imparting their own by contributing to those students’ experiences in the environment in which they’re learning. This is similar, in effect, to the compiler and test case feedback a student might get from a programming challenge, specifying logical or runtime errors. We characterise an instructivist approach, meanwhile, as one that focuses on instructing and specifying an understanding of the theories and problems. To do so, the tutor works with the student, on but outside the environment of the problem, and aims to build a specific understanding of the concepts (e.g., ‘assigning a variable twice in the same function causes the second assignment to override the first assignment’). In other words, they directly impart a specific model or understanding of a concept to the student. This is similar to the incrementality and specific instruction of many teaching materials, which present concepts explicitly and one by one. This gives rise to our first axis - whether feedback pertains to a specific instance of a student’s mistake (instance oriented) or to the theory or concepts that underly the error (theory oriented).

The second axis of our classification is whether the feedback is problem or solution oriented. We conceive of problem-oriented feedback (e.g., ‘your variable assignment is incorrect in line 4’) as more constructivist in nature – it provides information about the environment and leaves the learner to construct knowledge based on subsequent interactions without specifying an alternative and correct understanding. Meanwhile, we see solution-oriented feedback (e.g., ‘you can fix your variable assignment in line 4 by...’) as closer to instructivist, as it provides more direct instruction for the learner. This forms the second axis.

This gives rise to the four quadrants shown in Fig 1 – problem-instance, problem-theory, solution-instance and solution-theory. We have selected these features because they are readily observable, most feedback can be categorised according to these axes and due to their relationships with learning theories outlined above. We treat these classifications as categorical, although series of feedbacks could in aggregate be regarded as more or less aligned with either point of each axis according to their composition. It also enables two types of analyses that are important in creating a holistic understanding of feedback analysis – feedback sequencing and temporally variable learning effects.

Feedback sequencing refers to the order in which types of feedback are given. In environments where neither a constructivist nor an instructivist style is specified, it is often the case that multiple types of feedback will be employed within the same intervention. For example, an educator may give feedback ‘you need to concatenate your strings in line 4’ (solution-instance) and then explain that ‘strings in Python can be concatenated using the + symbol’ (solution-theory). It is useful to understand the sequence of feedback as a potential variable in understanding effectiveness. It also presents the opportunity for a non-binary analysis of these paradigms and allows us to question whether the combination of paradigms, as opposed to the singular selection of one, might be more effective. This empirical approach contrasts with theoretical analyses through which the two paradigms provide opposing conceptualisations of the learning process.

Temporally variable learning effects refers to the learner’s changing relationship with learned concepts over time. This is important because it speaks to one of the questions that divides practitioners of constructivist and instructivist teaching – that of goals. As Kapur [16] shows, methods for teaching that create high performance, meaning that they support the pursuit of goal-oriented behaviour in the short term such as passing the test cases in a coding challenge, can be unproductive because they do not promote long-term understanding or knowledge retention. The inverse is problematic as it creates frustration and inhibits motivation in students. We can thus apply this model to classify feedback and then follow the student’s progression in both the short term and the medium or long term, observing how their understanding changes over time. The question of what outcome is desirable is a matter of
the teacher’s goals for the student, and will depend on the practitioner, but this framework gives a method of analysis regardless. A related question is how performance changes in different types of tasks. As learners progress from simple to complicated to complex problem solving, the ability to apply knowledge in novel ways becomes important. The effect on this, too, can be tracked from the point feedback is given.

As an initial exploration into the application of this framework to analysis of feedback in digital learning environments, we present the results of a case-study on the topic. This case study serves to demonstrate an initial application of the taxonomy presented, with the aim of showing the types of challenges such an approach can help us tackle.

4 Case Study

We analysed an N=2908 set of messages between tutors and students from the first week of the Grok Learning NCSS Challenge 2019 Intermediate2 aligned to the Year 9/10 Australian federal Digital Technologies curriculum. Students participating spanned grades three to twelve in Australian primary and secondary schools. Each week, they are presented with a series of slides with background theory, based on which they complete coding exercises pertaining to concepts learned and building on previous concepts. In addition to the automated feedback generated by carefully designed test cases, students can also receive feedback from human tutors. These tutors are a mix of teachers, IT professionals and university students completing relevant degrees. They either ask a tutor for help through an inbuilt chat feature or were sometimes asked by tutors if they would like assistance if they made five consecutive failed attempts. The majority of the ~10,000 students did not make use of the tutoring functionality in that week and have not been included in this analysis. We have included every student who made use of the tutoring functionality in week one of the course. We discarded messages from students to tutors and messages sent after a successful student submission, leaving us with an N=1236 set of feedback.

For each student submission, Grok Learning evaluates code against a number of test-cases, testing different elements of the solution. For example, in a question about getting user input, the test cases might be: 1. Testing getting user input. 2. Testing capitalisation of user input prompt. 3. Testing whitespace in user input prompt. 4. Testing punctuation of user input prompt.

We consider a feedback block to be a series of feedbacks given that are not interrupted by a submission. Within a block, we disregard consecutive feedbacks of the same classification – we represent two consecutive problem-instance feedbacks within the same block as a single feedback. We call the sequence of feedback types in this block a feedback sequence. The length of the feedback sequence is the number of feedback encodings ascribed to that block (for example, a feedback sequence with feedback types solution-instance, solution-theory, problem-instance, solution-instance in that order has feedback sequence length four).

---

2 https://groklearning.com/challenge/
Fig 3 shows a strong tendency toward instance- and solution-oriented feedback. There are a few likely reasons for this. Firstly, it is natural for a tutor to help by talking about the problem facing the student at the time. This gives tutor and student a shared reference point for discussion and lends itself to instance-oriented feedback. Secondly, the content outlined within the platform is comprehensive and clear, and it is likely that this is generally seen as sufficient by tutors, who see less need to expound further on the topics covered. In support of this, a large amount of the theory-oriented feedback provided by tutors was in part or whole sending the students to reread content the tutor has identified as missing in students’ understanding. Thirdly, Grok Learning provides feedback of its own through test cases and the terminal, and this is generally problem-oriented (e.g., a runtime error is explicitly problem-oriented). As such, by the time tutors are engaged the student is likely to already have some problem-oriented feedback which has not sufficiently helped them, leading tutors to try a different approach.

We used two indicators to measure feedback effectiveness; the number of submissions students make on a question after receiving feedback before making progress (in other words, before passing more test-cases than they had before the feedback was given), and the length of the feedback sequence, as described previously. Longer feedback sequences, we reason, are likely to mean less clear and therefore less effective feedback.

Feedback with short-term effectiveness should result in the student requiring fewer subsequent submissions to progress. We define progression as passing at least one test-case that was not passed prior to the intervention. We tested this with each quadrant and along each axis and found no significant difference between any quadrant or pair of quadrants (Kruskal-Wallis test p-value = 0.46). Theory-oriented feedback resulted in slightly less submissions than instance-oriented (see Table 1 below), but a Mann-Whitney one sided test investigating this trend gave a p-value of 0.37, leading us to reject this as a meaningful effect. The same test investigating whether solution-oriented feedback resulted in less subsequent submissions returned a p-value of 0.10 and was also rejected. A substantial significant difference, however, was seen when looking at the percentage of students who failed to make progress and abandoned an exercise after receiving feedback. Students receiving problem-oriented (9.05%) and instance-oriented (11.63%) feedback were less likely to give up than those receiving solution-oriented (15.77%) or theory-oriented (17.78%) feedback.2 Population z-score tests affirmed this difference, with a p-value of 0.01 leading us to reject the null hypothesis that solution problem- and solution-oriented feedback had equal failure rates, and a p-value of 0.02 inferring the same of instance- and theory-oriented feedback.

Table 1: Subsequent student submissions to make progress after receiving each feedback type

<table>
<thead>
<tr>
<th>Feedback Type</th>
<th>( \bar{x} )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>problem</td>
<td>1.75</td>
<td>1.80</td>
</tr>
<tr>
<td>solution</td>
<td>1.77</td>
<td>1.46</td>
</tr>
<tr>
<td>instance</td>
<td>1.81</td>
<td>1.66</td>
</tr>
<tr>
<td>theory</td>
<td>1.69</td>
<td>1.38</td>
</tr>
</tbody>
</table>

The similarity of results suggests each approach to be roughly as effective in the short term as any other. This may be because tutors are adept at choosing approaches tailored to the individual’s needs. Contrastingly, failure rates show a much stronger distinction between approaches. The data shows that students who received feedback and passed all test cases submitted an average of 6.25 times per question, compared to the 4.09 times that students submitted who received feedback but gave up without passing all test-cases. As such, it may be that problem-oriented and instance-oriented feedback have a positive motivational impact, causing students to stick with an exercise for longer.

We investigated feedback sequence length by analysing feedback blocks. Students received few feedback blocks (\( \bar{x} = 1.19, \sigma = 0.53 \)) per exercise, and with 87.16% of exercise attempts by students receiving feedback eventually resulting in the student passing all tests, they were generally effective. Some variation still existed, and we theorised that the type of feedback that started a given block might be correlated with the feedback sequence length. A Kruskal-Wallis test on the four quadrants reported a p-value of 3.52E-5 leading us to reject the hypothesis that each quadrant has equal mean values. Analysing along the instance–theory axis produced minor effects (see Table 2), with a Mann-Whitney one sided test p-value of 0.29 suggesting that theory-oriented feedback is not likely to produce shorter feedback sequences that instance-oriented is. Along the problem–solution axis, the same test suggests
that solution-oriented feedback provided shorter sequences, with a p-value of 1.3E-5.

Table 2: Feedback block lengths based on orientation of first feedback in the block

<table>
<thead>
<tr>
<th></th>
<th>(\bar{x})</th>
<th>(\sigma)</th>
</tr>
</thead>
<tbody>
<tr>
<td>problem</td>
<td>1.96</td>
<td>1.79</td>
</tr>
<tr>
<td>solution</td>
<td>1.46</td>
<td>0.99</td>
</tr>
<tr>
<td>instance</td>
<td>1.60</td>
<td>1.31</td>
</tr>
<tr>
<td>theory</td>
<td>1.64</td>
<td>1.31</td>
</tr>
</tbody>
</table>

This makes intuitive sense, as solution-orientation seems a more direct approach than asking learners to derive a solution from a problem. Another explanation of this result, however, is that some tutors may be likely to respond reactively and instinctively to learners and pointing out a problem can be a more instinctive response than providing a solution-oriented approach. This would suggest that tutors whose feedback isn’t as well thought-out may provide more problem-oriented feedback, not necessarily that problem-oriented feedback is less effective. It also points to a potential problem with this indicator – it narrowly measures the student’s engagement with the tutor, not their subsequent attainment. In particular, it fails to account for any sort of medium to long term effects of the feedback – one can easily imagine a scenario in which a longer initial exchange contributes to a student having a better developed model of the content.

These two indicators seem to provide opposing results, which further illustrates the multivariate nature of effective feedback. These results support the idea that a comparison of different pedagogical approaches, such as instructivism and constructivism, must be done in relation to a specified target outcome. As such, it cements the idea that feedback approaches should be chosen based on the educator’s goals, and that, with a readily applicable taxonomy of feedback, one can measure how different factors influence attainment of different goals. This empowers the educator to select feedback approaches based on desired outcomes. As suggested earlier, we believe that early stage computer science education, particularly in digital environments, has reasonably well-defined goals. As such, this case study motivates investigation into the medium to long-term effects of these different feedback approaches. We aim to do so in the near future, by assigning test-cases labels according to the concept they test for, from the week of data used above and the subsequent four weeks of the challenge. By doing this, we can discern which concepts feedback was given in relation to and analyse students’ successive errors in those concepts across a five-week period.

5 Conclusions and Further Research

The shift towards digital learning environments presents a plethora of opportunities for educators and learners. These environments can be scalable, broadly accessible and effective in supporting learning outcomes. This is particularly true in computer science education, which has been on the vanguard of digitising teaching and learning. With the wealth of data collected in digital computer science learning environments, we have an opportunity to empirically understand how the feedback so important to achieving learning outcomes can be most effectively provided. This stands to increase the effectiveness of these platforms, and to contribute to the literature of pedagogical approaches more broadly. To do so, it is clear that we need a taxonomy of feedback, like the quadrants presented here, that is readily applicable in these environments. By analysing feedback through this lens, we can understand its role at a deeper level than might otherwise be possible.

Analysis of data within Grok Learning has demonstrated observable differences between types of feedback. Whilst little difference was observed between feedback types in the number of submissions required to make progress after an intervention, problem- and instance-oriented feedback was correlated with significantly lower student failures rates. Meanwhile, the length of feedback sequence was well correlated with the problem- or solution-orientation, with the latter aligned with smaller sequences. That these two indicators produced partially contradictory results is a testament to nuance and need for analysis within the topic of effective pedagogical feedback. We have suggested that the goals of an educational approach or environment are a key factor in determining what makes an ‘effective’ pedagogy. The taxonomy presented here, then, should be used to assess different feedback approaches against the goals of educational programs. In particular, the next research step is the application of this taxonomy to track medium- and long-term student outcomes post feedback – if digital environments are aiming to create long-term knowledge retention, that’s exactly what we should be testing.

Finally, we come back to the unification of learning science and learning analytics. Historically, the lack of empirical comparison between approaches such as instructivism and constructivism has been justified by a divergence of goals. Our goal here, then, has been to present a framework through which different approaches to feedback can be analysed empirically, so that more effective approaches can be chosen in response to educators’ goals across teaching disciplines. This method of analysis is perhaps most directly applicable to computer science education, with its data rich environments and developed digital learning environments and work here could underpin broader application. Our hope, however, is that it these methods can develop, reaching beyond that context to contribute to the canonical understanding of learning science. With the right analytical approaches, this digitisation is an opportunity to better understand the characteristics of a body of theories and frameworks of feedback, and learning more generally, through an empirical lens.

REFERENCES

Toward Empirical Analysis of Pedagogical Feedback in Computer Programming Learning Environments


