Understand students’ self-reflections through learning analytics

Citation for published version:

Digital Object Identifier (DOI):
10.1145/3170358.3170374

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
LAK ’18 Proceedings of the 8th International Conference on Learning Analytics and Knowledge

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Understand students’ self-reflections through learning analytics

Author one
Author one department
Author one institution
Author one city, Author one state
author.one.email@institution

Author two
Author two department
Author two institution
Author two city, Author two state
author.two.email@institution

Author three
Author three department
Author three institution
Author three city, Author three state
author.three.email@institution

Author four
Author four department
Author four institution
Author four city, Author four state
author.four.email@institution

Author five
Author five department
Author five institution
Author five city, Author five state
author.five.email@institution

Author six
Author six department
Author six institution
Author six city, Author six state
author.six.email@institution

Author seven
Author seven department
Author seven institution
Author seven city, Author seven state
author.seven.email@institution

ABSTRACT
Reflective writing has been widely recognized as one of the most effective activities for fostering students’ reflective and critical thinking. The analysis of students’ reflective writings has been the focus of many research studies. However, to date this has been typically a very labor-intensive manual process involving content analysis of student writings. With recent advancements in the field of learning analytics, there have been several attempts to use text analytics to examine student reflective writings. This paper presents the results of a study examining the use of theoretically-sound linguistic indicators of different psychological processes for the development of an analytics system for assessment of reflective writing. More precisely, we developed a random-forest classification system using linguistic indicators provided by the LIWC and Coh-Metrix tools. We also examined what particular indicators are representative of the different types of student reflective writings.

CCS CONCEPTS
• Information systems → Clustering and classification;
• Applied computing → E-learning; Distance learning;

KEYWORDS
learning analytics, text mining, self-reflections, online learning

ACM Reference format:
Author one, Author two, Author three, Author four, Author five, Author six, and Author seven. 2018. Understand students’ self-reflections through learning analytics. In Proceedings of 8th International Learning Analytics and Knowledge (LAK) Conference, Sydney, NSW, Australia, March 5–9 2018 (LAK’18), 10 pages.
https://doi.org/10.1145/2883851.2883950

1 INTRODUCTION
An important characteristic of modern education is the focus on developing student higher cognitive skills and critical thinking. In this regard, some of the most fundamental learning activities relate to the use of (self-)reflection. The act of reflection is widely considered to be the essence of thinking process [22]. Reflection also represents an integral part of student self-regulation, and is essential for metacognitive adaptation of study approaches and goals [69]. The benefits of reflection are well recognized in contemporary educational practice [62, 63].

Over the years, there have been many approaches developed for fostering student reflection [cf. 62]. Among the different strategies used, reflective writing represents one of the most popular methods. An approach that is widely used for triggering the process of self-reflection is that necessary for metacognitive regulation. Not surprisingly, the assessment of student reflective writings has been the focus of many studies. These studies have largely employed more manual and labour intensive content-analysis methods to evaluate student reflective writings [41]. With the recent advancements in the field of learning analytics, there have been some attempts
to use more automated methods. With learning analytics increasingly being used as an approach to promote student awareness and regulation of their learning activities [28], developing automated processes to assess and understand the content of student self-reflections is an important undertaking. For example, the work of Ullmann [62] provides early evidence how analytics methods can be used to understand reflection as expressed in student essays.

This present study examined the use of automated text analytics methods for assessing the content of students’ reflections on their learning activities. More precisely, we developed a learning analytics system for assessing reflection as expressed in student video annotations of their musical performances and examined what characteristics of their written language is most predictive of the different types of reflection.

2 BACKGROUND WORK

2.1 Reflection and Self-Regulated Learning

Across a range of disciplines, self-reflection is a key skill and strategy for students to cultivate as they enhance their higher order thinking skills and prepare for professional practice [19]. Reflection, or reflective practice, provides students with the opportunity to develop autonomy and confidence in their learning as they establish learning goals and take ownership of their learning strategies [14]. The higher education environment provides an opportune time for students to learn how to think independently, comment critically, and reflect on their learning [21] as they build the self-monitoring and self-regulating skills needed to be life-long learners [34]. Particularly, in clinical [36] or performance based disciplines [43] where students can watch a video recording of themselves demonstrating a particular skill, self-reflection exercises and assessment tasks can promote student self-assessment of their performance and identify areas of improvement. Reflective journals [14] and video annotation tools [3, 35] have shown to be effective tools for supporting students’ self-reflection and establishment of learning goals.

Despite the promotion of self-reflection skills through scaffolded activities and assessment, the depth of students’ reflection and the progression from potentially superficial levels of reflection (e.g., descriptive) to more higher-order and goal-oriented [8, 35] requires an analysis of the specificity or type of statements made. An examination of the type of reflection in varying pedagogical or instructional conditions, can help identify students who are struggling to grasp higher levels of reflective thought (e.g., establishing goals) and those who may be largely focused on describing their skill or performance rather than critiquing it further. Reflection that involves goal-setting is much more challenging [58] and strategies are required to support students mastery of it. For example, × × × [4], examined the effect of students’ experience with reflective tasks and the instructional conditions (graded vs ungraded activity) on the level or specificity of students’ reflective statements. Their study concluded that prior experience with reflection along with continual grades and formative feedback on their reflective tasks encourages a greater amount of higher order critical reflection (e.g, goal-orientated or analysis of their motive or effect of their performance). Hence, scaffolding and an early introduction of reflective practice in the curriculum is needed for raising the depth and complexity of student reflection [19].

2.2 Automated analysis of self-reflections

While analysis of student self-reflections provides important insights into the development of students’ higher order thinking, it is for the most part, very time-consuming manual process [61, 64, 65]. In most cases, it involves the quantitative content analysis [41] of student writings using a pre-defined coding scheme that focuses on identifying word indicators of the different facets of reflections [61]. Broadly speaking, different content analysis approaches exploit the underlying differences in the distributions of different linguistic categories between reflective and non-reflective statements and texts [63]. The majority of prior work has focused on the analysis of student essays and journal writings, with an emphasis on the depth of student reflection expressed (e.g., no reflection, simple reflection, and critical reflection) [66]. Not surprisingly, (self-)reflection in student writings was found to be substantially less frequent than desired [66], primarily on the descriptive [33] and shallow [56] levels.

Given the potential of computational methods for understanding student self-reflections, there have been several attempts to develop automated systems for assessment of student writings, including self-reflective texts. According to Ullmann [64], the existing automated content analysis systems can be divided into three broad and overlapping groups based on the adopted methodology:

1) Dictionary-based approaches [e.g., 15, 18, 46, 47, 61, 62],
2) Rule-based approaches [e.g., 30, 61, 65], and
3) Machine learning approaches [e.g., 1, 2, 5, 17, 46, 47, 62].

These three general approaches are also often combined. For example, Ullmann [61] proposed a system for identification of reflection in student essays using the combination of predefined dictionaries, regular expressions and rule-based analytics. Ullmann [61] also used synonym expansions to extend the list of words associated with reflective writings and provide more generalizable and stable performance. Similarly, Ullmann et al. [65] developed a rule-based system for reflection analysis in students’ blog postings using WordNet [25], Linguistic Inquiry and Word Count (LIWC) tool [59], Stanford NLP parser [45], and synonym database. Using a custom-built vocabulary of the important keywords and focusing on the type of pronouns used (e.g., first person singular, third person plural) Ullmann et al. [65] devised a set of rules for identification of the different elements of reflective writings. A similar approach based on LIWC [59] and Coh-Metrix [31, 49] has been utilized by × × × [2] for the identification of students’ level of critical thinking as expressed in discussion forum postings.

A further common approach to analyzing student writings is based on the use of natural language processing (NLP) methods. This is often applied in combination with different machine learning algorithms. The simplest NLP methods use frequencies of N-grams (i.e., word sequences of length N) as classification features [e.g., 1, 62]. For example, Ullmann [62] used N-grams as features for binary classification of 5,081 student reflection sentences and elements of reflective writings (i.e., experience, feelings, personal, critical stance, perspective, outcome), reporting classification accuracy as Cohen’s χ range of .49–.83, depending on the particular coding category. Similarly, Gibson et al. [30] used part-of-speech (POS) tagging to match students’ writings to the common POS phrases indicative of student’s metacognitive activities, while Latent Semantic Analysis
(LSA) has been used by Cheng [17] to understand reflection in English language learners. Likewise, the Gibson and Kitto [29] NLP-based approach utilized the TF-IDF scoring [40], Latent Dirichlet Allocation (LDA) [10], and different keyword-based metrics for identification of the level of subjectivity and affectivity in students’ reflective writings. Finally, × × × [1] used POS-tagging. Name-entity tagging, and syntactic dependency parsing (via Stanford NLP toolkit [45]) to build a classification system for examining students’ levels of critical thinking.

3 RESEARCH QUESTIONS

While there has been a substantial amount of research on automated assessment of student reflective writings, the primary domain of analysis were long, complex texts, such as essays, blogs, or journals, in which students were expected to exercise reflective and critical thinking. As reflection is typically represented in just a small part of the written text, a large part of the existing research focused on the identification of different parts of written text that represent different types and facets of reflection. This was typically achieved through a combination of custom-built keyword and phrase matching mechanism, or by a data-driven NLP indicators, such as N-grams, that were chosen depending on the specifics of a particular study context. Hence, there are concerns regarding the external validity in the literature published up to date, with regards to what are the highly predictive – and psychologically sound – indicators of (self-)reflection in student writings and how they can be used to develop analytics systems for reflection assessment. As such, the research questions addressed in this study are

Research Question 1:
What are the linguistic indicators of self-reflection, as captured in students’ writings?

Research Question 2:
Can the identified indicators of self-reflection be used to develop an automated system for assessment of students’ self-reflection?

To address these questions, we used psychologically-sound and well-established linguistic measures of different psychological processes (e.g., affective, cognitive, social, biological) provided by the widely used LIWC [59] and Coh-Metrix [31, 49] tools in addition to the widely used N-grams, in order to develop an automated classification system for reflection assessment. To make the identification of the relevant reflection indicators more precise, we focused on analyzing short self-reflective writings rather than longer (e.g., essays or blogs) texts which typically contain much lower proportion of reflective writing. In particular, we examined students’ self-reflection in their short annotations of the video recordings of their own musical performances.

4 METHOD

4.1 Study data

4.1.1 Study setting. The dataset in the present study is the same dataset that was used in the study described in [3]. The data comes from the four undergraduate courses in performing arts discipline offered in the 2012/2013 academic year at a large research-intensive public university in Canada. Course 1 and Course 2 were offered in the Fall 2012 semester while Course 3 and Course 4 were offered in the Winter 2013 semester. In all four courses, students were providing self-reflections on the video recordings of their own musical performances. In Course 1, the recordings were of students’ group performances while in the other three courses, video recordings were of students’ individual performances (Table 1). In addition, in Course 1 and Course 4, the creation of self-reflections was optional activity, while it was a course requirement in Course 3 and Course 4 and part of student assessment. In total, there were 77 different students across the four courses, with some students taking more than one course.

To create their self-reflections, students used Collaborative Lecture Annotation System (CLAS) [50, 55], which is a software tool that enables students to annotate video materials, which are in this case videos of their art performances. In terms of the functionality, CLAS enables students to create time-stamped annotations, which are associated with a particular part of a video, and general annotations which are not associated with any part of the video and used to create general comment or summary of the video. Both time-stamped and general annotations can be either private or public, with the latter providing the opportunity for student collaboration and peer feedback.

4.1.2 Content analysis. After students’ self-reflections were collected, the quantitative content analysis [41] was undertaken to categorize each student reflection using the coding scheme adapted

<table>
<thead>
<tr>
<th>Table 1: Description of included courses and coded units of analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>Course 1</td>
</tr>
<tr>
<td>Course 2</td>
</tr>
<tr>
<td>Course 3</td>
</tr>
<tr>
<td>Course 4</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Description of coding categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Observation</td>
</tr>
<tr>
<td>Motive</td>
</tr>
<tr>
<td>Goal</td>
</tr>
</tbody>
</table>
As the first step in our analysis process, we first split the data into training and test datasets. The distribution of different codes is shown in Table 3. We see that the majority of units were coded as either goal indications (55.63%), expressions of any of the three reflection types, and it was used to code 4.74% of the analysis units. Other also included the category Goal.

In total, training and test datasets contained 3,322 and 1,108 instances, respectively (Table 3). It should be noted that training and test datasets are created in a stratified manner, which means that the original proportions of coding categories (i.e., Observation, Goal, Motive, and Other) is preserved in both subsets (Table 3).

### 4.2 Training and test data preparation

As the first step in our analysis process, we first split the data into training and test datasets (75% and 25% of the whole corpus, respectively), as commonly done in the machine learning [32, 54]. The model development and parameter tuning are done using the training set, while the final evaluation of model’s performance is done on test set. By doing this, we prevent for overestimating the model performance which will occur if we estimated the model accuracy on test set. By doing this, we prevent for overestimating the model performance which will occur if we estimated the model accuracy on test set. By doing this, we prevent for overestimating the model performance which will occur if we estimated the model accuracy on test set. By doing this, we prevent for overestimating the model performance which will occur if we estimated the model accuracy on test set.

### 4.3 Feature extraction

In order to develop a classification system for student reflections, we extracted several different types of features. The extracted features were heavily based on the existing work in educational text and discourse analysis [e.g., 1, 2, 5, 6, 20, 23, 24, 31, 37, 49, 57, 60, 67], including the features which are strongly theory-driven and empirically validated. In total, we extracted 503 different features which we describe in the reminder of this section.

#### 4.3.1 N-grams

As commonly done in text classification systems, we extracted basic N-grams features (i.e., unigrams, bigrams, and trigrams) from the training data (i.e., 75% of the whole corpus). Prior to N-gram extraction, we first removed stopwords, which are the highly frequent words in English (e.g., a, the, be, can, have) that do carry useful information for classification purposes [54]. Given that the use of N-grams results in inflation of the feature space and overfitting of the training data, we extracted only top 100 unigrams, bigrams, and trigrams to keep the size of the feature space limited and less prone to overfitting. The top ten most frequent unigrams, bigrams, and trigrams (Table 4) are about the quality of student performances and students’ needs, goals, and feelings which could be used to gauge the type of student reflection. As expected, we also see a sharp decline in N-gram frequencies as N increases.

After we extracted a set of 300 N-gram features from the training set, we extracted the same set of N-grams features from the test set (i.e., the remaining 25% of the whole corpus that were not included in the training set). Therefore, the definition of the feature space only depends on the training data, while the test data is completely put aside and used only for the final validation of the classifier performance.

#### 4.3.2 LIWC features

In addition to N-gram features, following the work of [2], we used the Linguistic Inquiry and Word Count (LIWC) tool [59] to extract a large set of linguistic measures which are indicative of a large set of biological and psychological processes (e.g., perceptual, cognitive, affective, social), as well as different topics (e.g., work, achievement, personal, leisure, time) and linguistic categories (e.g., nouns, verbs, adjectives). The previous work [2] indicated that LIWC measures can be successfully used within learning analytics systems to uncover important psychological processes behind student behavior observed in trace data logs. In the current study, we used the 2015 version of the LIWC tool which provides the total of 93 empirically validated linguistic measures [cf. 59], including four high-level measures: (1) analytical thinking, (2) social status, leadership, and confidence, (3) authenticity, and (4) emotional tone.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>need</td>
<td>383</td>
<td>left hand</td>
<td>112</td>
<td>practice front mirror</td>
<td>17</td>
</tr>
<tr>
<td>conducting</td>
<td>279</td>
<td>eye contact</td>
<td>71</td>
<td>use left hand</td>
<td>14</td>
</tr>
<tr>
<td>think</td>
<td>248</td>
<td>need work</td>
<td>55</td>
<td>third goal would</td>
<td>11</td>
</tr>
<tr>
<td>music</td>
<td>239</td>
<td>make sure</td>
<td>54</td>
<td>make eye contact</td>
<td>10</td>
</tr>
<tr>
<td>really</td>
<td>200</td>
<td>front mirror</td>
<td>36</td>
<td>second goal would</td>
<td>10</td>
</tr>
<tr>
<td>hand</td>
<td>182</td>
<td>goal would</td>
<td>32</td>
<td>first goal would</td>
<td>10</td>
</tr>
<tr>
<td>practice</td>
<td>181</td>
<td>feel like</td>
<td>30</td>
<td>three critical goals</td>
<td>8</td>
</tr>
<tr>
<td>ensemble</td>
<td>171</td>
<td>beat pattern</td>
<td>30</td>
<td>critical goals improvement</td>
<td>8</td>
</tr>
<tr>
<td>work</td>
<td>170</td>
<td>right hand</td>
<td>29</td>
<td>really need work</td>
<td>8</td>
</tr>
<tr>
<td>beat</td>
<td>161</td>
<td>also need</td>
<td>26</td>
<td>influence sound moment</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 3: Distribution of coding categories in test and train data sets

<table>
<thead>
<tr>
<th>Category</th>
<th>Dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train (%)</td>
<td>Test (%)</td>
<td>All (%)</td>
<td></td>
</tr>
<tr>
<td>Observation</td>
<td>1,135 (34.17%)</td>
<td>382 (34.48%)</td>
<td>1,517 (34.24%)</td>
<td></td>
</tr>
<tr>
<td>Goal</td>
<td>1,848 (55.63%)</td>
<td>625 (56.41%)</td>
<td>2,473 (55.82%)</td>
<td></td>
</tr>
<tr>
<td>Motive</td>
<td>174 (5.24%)</td>
<td>56 (5.05%)</td>
<td>230 (5.19%)</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>165 (4.97%)</td>
<td>45 (4.06%)</td>
<td>210 (4.74%)</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3,322 (100%)</td>
<td>1,108 (100%)</td>
<td>4,430 (100%)</td>
<td></td>
</tr>
</tbody>
</table>
Understand students’ self-reflections through learning analytics

4.3.3 Coh-Metrix features. In addition to LIWC, similarly to $\times \times \times$ [2], we also used the Coh-Metrix tool [31, 49], which is a text analytics tool designed to measure different aspects of writing cohesion. Coh-Metrix provides 109 different measures of text cohesion (i.e., referential, causal, co-reference, temporal, spatial, and structural cohesion), several measures of text complexity and readability, and measures of linguistic category use.

Coh-Metrix has been extensively used in many studies in the domain of collaborative learning to assess student outcomes [23], online discourse [24, 68], development of social ties [37–39], quality of student essays [7, 48], and learning resources [31]. Coh-Metrix has also been successfully used in learning analytics systems for assessing student-produced writings, such as student discussion messages [2, 67]. Given the goal of understanding the processes driving student self-reflections, Coh-Metrix provides a valuable set of empirical measures that can be used to understand the characteristics of each of the types of student reflections.

4.3.4 Context features. Given that several units of analysis can be present in a single sentence, we also included a single binary feature first_in_sentence which captures whether a particular unit of analysis is the first (or the only) unit in a given sentence. We hypothesized that students’ observations would more often be first in a sequence of annotations given their sensemaking nature.

4.4 Data preprocessing

After feature extraction, we addressed the problem of class imbalance, as visible in Table 2. Following the approach suggested by $\times \times \times$ [2], we used the Synthetic Minority Oversampling Technique (SMOTE) [13, 16], which is a popular method for addressing the class imbalance problem. The SMOTE algorithm works by constructing additional synthetic data points as a linear combination of the existing data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.

To increase the size of the minority class by $N$ times, each minority-class data point would be processed $N$ times. In contrast, to increase the size of the minority class by less than 100%, first a subset of the original data points was selected and then each of those data points would be processed exactly once. Figure 1 illustrates the application of SMOTE algorithm in our training set. The size of Other category was increased 10-fold (from 1135 to 1816) by first selecting 60% of the original data points. To process an existing data point $X$ in an $n$-dimensional feature space $X = (x_1, x_2, x_3, \ldots, x_n)$ using the SMOTE algorithm:

- Find $K$ nearest neighbors of $X$ (in our case, $K=5$) belonging to the same minority class.
- Select at random one of those $K$ nearest neighbors (called $Y$).
- Generate a new synthetic data point as a random linear combination of $X$ and $Y$:

$$Z = X + c \times (Y - X)$$

where $c$ is a random number between 0 and 1.
of trees in ensemble, and (2) mtry: the number of random features used by each tree. The number of trees in the ensemble should be sufficiently large so that the performance of the classifier is stabilized [51] while the number of features used by each tree should be carefully optimized to balance bias-variance tradeoff [32]. According to Oshiro et al. [51], ensembles of 64–128 trees are recommended to balance between the processing time, memory usage, and classification accuracy. This recommendation is aligned with our previous implementations of random forests [2] where the classification performance stabilized from around 100–150 trees. Still, given the relatively small size of our training set (7219 instances), the processing time and memory constraints were less critical so we decided to use 500 trees in the ensemble (i.e., ntree = 500).

Finally, to optimize mtry parameter, we used ten repetitions of 10-fold cross-validation to examined 19 candidate values: 2, 3, 4, 6, 8, 11, 15, 20, 27, 36, 48, 65, 87, 156, 209, 279, 373, and 500. The actual parameter values were generated by the caret package and its default grid search strategy.

### 4.6 Implementations

The implementation of the classifier was done in the Python and R programming languages and by using several software packages and libraries:

1. The extraction of N-grams was done using NLTK library [9] for Python programming language.
2. The extraction of psychological indicators was done with LIWC 2015 tool [53, 59].
3. The extraction of text coherence measures was performed with Coh-Metrix toolkit [31, 49].
4. Stratified sub-sampling of test and train data was done through scikit-learn [52] machine learning library for Python programming language.
5. The development of a random forest classifier was done using randomForest R package [44], and finally,
6. The model training, selection, and validation was performed with caret R package [27].

### 4.7 Limitations and future work

The major limitation of the adopted approach is that the collected data are from the same domain (i.e. performing arts) and thus, might not be representative of a broader range of student self-reflections across different disciplines. As such, one direction for our future work will be to examine the performance of the developed classification scheme on datasets from different study domains. Moreover, there are several potentially useful classification features which have not been included in the design of our system. For example, it is likely that the inclusion of the codes from student’s previous reflections as classification features would provide important additional information that would substantially increase the classification accuracy. For example, if a student wrote an observation reflection, then it is more likely that his following reflection would be goal or motive reflection. However, at the moment, each annotation is categorized in isolation from all other reflections made by a student, which likely reduces the classifier’s performance. In this regard, the use of structured classification approach, such as one employed by [67] is an important direction for the future work.

### 5 RESULTS

#### 5.1 Model training and evaluation

Figure 2 and Table 5 show the results of random forest model optimization via cross-validation. The best performance of .89 classification accuracy (SD = .01) and Cohen’s κ of .85 (SD = .02) was achieved with 116 features per decision tree on the training dataset. The difference between the worst- and best-performing model was 0.07 in classification accuracy and .10 κ which confirms the importance of parameter optimization and tuning on the final model performance (Table 5). The performance of the random forest model on the complete training set using the optimal mtry value is shown on Figure 3. We can see that the performance of the classifier stabilized with around 100 decision trees, indicating that 500 trees selected was more than enough to ensure good classifier performance. The average out-of-bag (OOB) error rate was .12, suggesting only 12% of the data points being misclassified in the training set. As expected, the error rates for the two most resampled classes (i.e., Other and Motive) were the lowest, while the highest error rate was observed for Observation category which was not resampled.

After developing the random classifier on the training data, we validated its performance on the holdout test data (25% of the whole

<table>
<thead>
<tr>
<th>mtry</th>
<th>Accuracy</th>
<th>Cohen’s κ</th>
<th>mtry</th>
<th>Accuracy</th>
<th>Cohen’s κ</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>.81 (.01)</td>
<td>.75 (.02)</td>
<td>48</td>
<td>.88 (.01)</td>
<td>.85 (.01)</td>
</tr>
<tr>
<td>3</td>
<td>.84 (.01)</td>
<td>.79 (.02)</td>
<td>65</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
</tr>
<tr>
<td>4</td>
<td>.86 (.01)</td>
<td>.81 (.02)</td>
<td>87</td>
<td>.88 (.01)</td>
<td>.84 (.02)</td>
</tr>
<tr>
<td>6</td>
<td>.87 (.01)</td>
<td>.83 (.01)</td>
<td>116</td>
<td>.89 (.01)</td>
<td>.85 (.02)</td>
</tr>
<tr>
<td>8</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
<td>156</td>
<td>.88 (.01)</td>
<td>.84 (.02)</td>
</tr>
<tr>
<td>11</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
<td>209</td>
<td>.88 (.01)</td>
<td>.84 (.02)</td>
</tr>
<tr>
<td>15</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
<td>279</td>
<td>.88 (.01)</td>
<td>.84 (.02)</td>
</tr>
<tr>
<td>20</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
<td>373</td>
<td>.87 (.01)</td>
<td>.83 (.02)</td>
</tr>
<tr>
<td>27</td>
<td>.88 (.01)</td>
<td>.84 (.01)</td>
<td>500</td>
<td>.86 (.01)</td>
<td>.82 (.02)</td>
</tr>
<tr>
<td>36</td>
<td>.88 (.01)</td>
<td>.85 (.01)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Min: .81 .75 Range: .07 .10 Mean: .87 .83

![Figure 2: Random forest parameter tuning results.](image-url)
dataset). Our random forest classifier achieved .75 classification accuracy (95% CI[0.72, 0.77]) and Cohen’s κ of 0.51 which is considered “Moderate” accuracy above the pure chance level [42]. The confusion matrix for the test data is shown in Table 7. We see that error rate for the Goal category is the lowest, followed by the moderate error rate for the Observation category. In contrast, we see that the Other and Motive categories were mostly misclassified as belonging to two former large categories.

Finally, to examine the value of the SMOTE preprocessing, we examined the confusion matrix of the random forest model developed using the original training and test datasets. The optimal mtry value was 500 by which the classifier obtained .73 (SD = .02) classification accuracy and Cohen’s κ of .48 (SD = .04). Further validation of the classifier performance on the holdout test data showed .74 classification accuracy (95% CI[.72, .77]) and Cohen’s κ of 0.50 which was slightly lower than the classifier performance obtained after the SMOTE pre-processing.

5.2 Feature importance analysis

In addition to assessing the classification accuracy, we also examined the contribution of different features to random forest performance. Table 8 provides the summary of feature MDG scores, while Figure 4 shows MDG scores for all 500 classification features.

Table 6: Train data confusion matrix for the final model

<table>
<thead>
<tr>
<th>Actual</th>
<th>Other</th>
<th>Observation</th>
<th>Goal</th>
<th>Motive</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>1686</td>
<td>80</td>
<td>49</td>
<td>0</td>
<td>.07</td>
</tr>
<tr>
<td>Observation</td>
<td>5</td>
<td>1655</td>
<td>177</td>
<td>11</td>
<td>.10</td>
</tr>
<tr>
<td>Goal</td>
<td>6</td>
<td>373</td>
<td>1422</td>
<td>15</td>
<td>.22</td>
</tr>
<tr>
<td>Motive</td>
<td>1</td>
<td>59</td>
<td>68</td>
<td>1612</td>
<td>.07</td>
</tr>
</tbody>
</table>

Table 7: Test data confusion matrix for the final model

<table>
<thead>
<tr>
<th>Actual</th>
<th>Other</th>
<th>Observation</th>
<th>Goal</th>
<th>Motive</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>9</td>
<td>9</td>
<td>27</td>
<td>0</td>
<td>.80</td>
</tr>
<tr>
<td>Observation</td>
<td>1</td>
<td>250</td>
<td>131</td>
<td>0</td>
<td>.34</td>
</tr>
<tr>
<td>Goal</td>
<td>1</td>
<td>59</td>
<td>564</td>
<td>1</td>
<td>.10</td>
</tr>
<tr>
<td>Motive</td>
<td>0</td>
<td>30</td>
<td>25</td>
<td>1</td>
<td>.98</td>
</tr>
</tbody>
</table>

We see a wide spread in MDG scores; 50% of features obtained an MDG score below 1.06 and 75% of features obtained an MDG score below 15.34. In contrast, certain features obtained much higher MDG scores, with the maximum MDG score of 219.94.

The detailed analysis of top twenty most important classification features is given in Table 9. While 146 classification features had above average MDG scores, given the space limitations, we focused our analysis on top twenty. We see that the most important classification feature was the LIWC category of perceptual words (liwc. see). In addition the use of past-oriented words (liwc. focuspast), punctuation, causal words, passive voice, and connectives were among the most important classification features.

Among the Coh-Metrix features, the most important were the ratio of causal particles to causal verbs (cm. SMCAUSr), use of agentless passive voice (cm. DRPVAL), use of nouns (cm. WRDNOUN) and noun phrases (cm. DRPPN), use of connectives (cm. CNCCaus), causal verbs (cm. SMCAUSv), intentional cohesion of the text (cm. SMINTER), and number of words before main verbs in sentences (cm. SYNLE).

The Motive reflections had the highest number of causal particles to causal verbs and words before the main verbs, indicating the complex language structure used to describe student motivation. Similarly, the use of agentless passive voice and connectives was strongly associated with the Motive category that also exhibited the highest intentional cohesion. In contrast, highest numbers of nouns and noun phrases were associated with the Other category, whereas causal verbs were most strongly associated with Goal category.

The most important LIWC features were related to students use of perceptual words (liwc. see) which were most strongly associated with the Observation and Goal categories and the least with the Other category. The Observation and Motive reflections also had a strong focus on the past events (liwc. focuspast), whereas Goal reflections did not. Somewhat unexpectedly, words related to

Table 8: Summary of classification feature importance

<table>
<thead>
<tr>
<th>Min.</th>
<th>Q1</th>
<th>Median</th>
<th>Mean</th>
<th>Q3</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.12</td>
<td>1.06</td>
<td>10.82</td>
<td>15.34</td>
<td>219.94</td>
</tr>
</tbody>
</table>

Figure 4: Feature importance by Mean Decrease Gini (MDG) measure. Dotted blue line shows median MDG score (1.06), while solid blue line shows average MDG score (10.81).
The classification results on the testing dataset showed that the use of N-grams and LIWC and Coh-Metrix features provides a good basis for the development of an automated self-reflection classification system. Cohen’s κ of 0.51 represents a moderate level of agreement above the change level [42]. These results are promising and showing the potential of our approach. The results also indicate the significant benefits of classifier parameter tuning, given the substantial variation in the classifier’s performance on the training dataset (Table 5). Table 5 indicates that 7% of the classification accuracy and .10 Cohen’s κ can be solely attributed to the optimization of the mtry parameter (i.e., the number of attributes used in each tree of the forest). The most directly comparable results are by Ullmann [62] who reported slightly higher Cohen’s κ values (.49–.83), albeit on a different, binary classification problem with different coding categories.

A further contribution from the study is the examination of the important classification features. While SVMs provided the best performance in most experiments by Ullmann [62], we opted for more interpretable classification methods which can be used to improve conceptual understanding of students’ self-reflection. Our results showed that a small subset of highly predictive indicators can be used to distinguish between the different types of reflective statements (Table 9). In particular, several of the indicators that capture different linguistic structures (e.g., agentless passive voice density, syntactic pattern density, connectives) were identified as some of the best predictors of student self-reflection. Hence, in our future work, we will also examine the inclusion of syntactic dependency features, such as the ones used by x × x × [1].

The important classification indicators (Table 9) indicate they are for the well aligned with the previous research on student self-reflection. Both Observation and Motive showed the strong use of past-oriented words, which is not surprising given that both categories relate to the descriptions of previous events (i.e., their past

### Table 9: Twenty most important features and their mean scores for messages in different phases of cognitive presence

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>MDG</th>
<th>Other</th>
<th>Observation</th>
<th>Goal</th>
<th>Motive</th>
</tr>
</thead>
<tbody>
<tr>
<td>liwc.see</td>
<td>Perceptual processes: seeing (e.g., view, saw, seen)</td>
<td>219.94</td>
<td>1.01</td>
<td>1.64 (4.43)</td>
<td>1.62 (4.57)</td>
<td>1.55 (3.29)</td>
</tr>
<tr>
<td>cm.SMCAUSr</td>
<td>Situation model: ratio of casual particles to causal verbs</td>
<td>201.33</td>
<td>0.12</td>
<td>0.13 (0.34)</td>
<td>0.11 (0.30)</td>
<td>0.37 (0.47)</td>
</tr>
<tr>
<td>cm.DRNP</td>
<td>Syntactic pattern density: agentless passive voice density, incidence</td>
<td>183.90</td>
<td>2.46</td>
<td>3.02 (17.57)</td>
<td>2.03 (13.36)</td>
<td>4.66 (20.52)</td>
</tr>
<tr>
<td>cm.focusepast</td>
<td>Time orientation: focus towards past (e.g., ago, did, talked)</td>
<td>151.83</td>
<td>1.46</td>
<td>4.57 (6.34)</td>
<td>0.80 (2.81)</td>
<td>4.80 (6.66)</td>
</tr>
<tr>
<td>cm.WRDNOUN</td>
<td>Word information: noun incidence</td>
<td>120.80</td>
<td>252.12</td>
<td>186.47 (99.92)</td>
<td>208.13 (127.82)</td>
<td>194.34 (95.04)</td>
</tr>
<tr>
<td>liwc.ingest</td>
<td>Biological processes: ingestion (e.g., dish, eat, pizza)</td>
<td>112.30</td>
<td>0.58</td>
<td>0.30 (1.68)</td>
<td>0.33 (2.21)</td>
<td>0.39 (1.66)</td>
</tr>
<tr>
<td>cm.CNCaus</td>
<td>Connectives: causal connectives, incidence</td>
<td>105.42</td>
<td>28.27</td>
<td>20.86 (42.82)</td>
<td>28.57 (47.68)</td>
<td>42.91 (50.47)</td>
</tr>
<tr>
<td>trust.ensemble</td>
<td>Frequency of &quot;trust ensemble&quot; bigram</td>
<td>95.18</td>
<td>0.00</td>
<td>0.00 (0.04)</td>
<td>0.00 (0.05)</td>
<td>0.00 (0.00)</td>
</tr>
<tr>
<td>cm.SMINTER</td>
<td>Intentional cohesion: ratio of intentional particles to intentional actions/events</td>
<td>88.05</td>
<td>0.30 (0.66)</td>
<td>0.23 (0.49)</td>
<td>0.36 (0.59)</td>
<td>0.47 (0.68)</td>
</tr>
<tr>
<td>liwc.Period</td>
<td>Punctuation: use of full stop</td>
<td>75.64</td>
<td>9.17</td>
<td>6.38 (6.93)</td>
<td>7.14 (8.60)</td>
<td>5.45 (6.08)</td>
</tr>
<tr>
<td>cm.DRNP</td>
<td>Syntactic pattern density: incidence score of noun phrases</td>
<td>71.76</td>
<td>390.98</td>
<td>319.76 (115.17)</td>
<td>306.22 (133.34)</td>
<td>314.47 (105.34)</td>
</tr>
<tr>
<td>chamber.musict</td>
<td>Frequency of &quot;chamber music&quot; bigram</td>
<td>70.59</td>
<td>0.00</td>
<td>0.00 (0.08)</td>
<td>0.00 (0.03)</td>
<td>0.01 (0.09)</td>
</tr>
<tr>
<td>liwc.AllPunc</td>
<td>Punctuation: all (e.g., periods, commas, question marks)</td>
<td>66.26</td>
<td>19.22</td>
<td>13.02 (9.85)</td>
<td>14.06 (12.39)</td>
<td>11.66 (8.12)</td>
</tr>
<tr>
<td>liwc.cause</td>
<td>Cognitive processes: causality (e.g., because, effect)</td>
<td>62.82</td>
<td>2.02</td>
<td>1.85 (4.32)</td>
<td>2.03 (4.66)</td>
<td>3.98 (5.33)</td>
</tr>
<tr>
<td>cm.SMCAUSv</td>
<td>Situational model: incidence score of causal verbs</td>
<td>61.33</td>
<td>33.60</td>
<td>46.08 (59.57)</td>
<td>65.65 (76.14)</td>
<td>44.28 (49.62)</td>
</tr>
<tr>
<td>liwc.insight</td>
<td>Cognitive processes: insight (e.g., think, know)</td>
<td>59.62</td>
<td>2.98</td>
<td>3.64 (5.34)</td>
<td>2.36 (5.34)</td>
<td>3.50 (4.56)</td>
</tr>
<tr>
<td>cm.SYNLE</td>
<td>Syntactic complexity: mean number of words before the main verb in the main clause</td>
<td>57.85</td>
<td>2.36</td>
<td>2.65 (3.08)</td>
<td>1.78 (2.60)</td>
<td>3.53 (4.16)</td>
</tr>
<tr>
<td>liwc.home</td>
<td>Personal concerns: home (e.g., kitchen, landlord)</td>
<td>55.83</td>
<td>0.10</td>
<td>0.08 (0.85)</td>
<td>0.05 (0.59)</td>
<td>0.04 (0.49)</td>
</tr>
<tr>
<td>liwc.Analytic</td>
<td>Summary measures: the measure of formal, logical, and hierarchical thinking processes</td>
<td>52.24</td>
<td>60.02</td>
<td>54.60 (35.94)</td>
<td>70.94 (33.05)</td>
<td>57.36 (34.69)</td>
</tr>
<tr>
<td>liwc.percept</td>
<td>Perceptual processes: all (e.g., look, heard, feeling)</td>
<td>51.97</td>
<td>4.20</td>
<td>4.85 (6.80)</td>
<td>4.69 (7.31)</td>
<td>5.04 (5.67)</td>
</tr>
</tbody>
</table>

1 Mean decrease Gini impurity index.

### Notes

1. first_in_sentence was coded as: Yes=2, No=1
Understand students’ self-reflections through learning analytics

performance currently watched) and already identified by Ullmann [64] and Ullmann [63]. However, it is likely that the use of video recordings as a media to facilitate reflection had an impact on the use of perceptual words. If students were reflecting based on their memories of the past events, it is likely that they will use less perceptual words. As such, it seems important to provide students with not only instructional scaffolds, but also resources and materials for reflection in a format that will best promote (self)-reflection and critical thinking development.

We also see a strong use of words describing cognitive process of insight in the Observation and Motive categories. This is not surprising, given that reflection is one of the most effective approaches to fostering students’ higher order thinking skills which is conditioned upon inquiry and insightful thinking [11]. The Motive category was also associated with higher intentional cohesion and causality, which is well aligned with the properties of the Motive-Effect type of reflective statements that capture student intentions and outcomes of particular actions. Our results also provide more detailed insights into the particular syntactic structures used to express motives and effects of student actions. We see that Motive category is associated with more agentless passive statements, higher use of connectives, higher ratio of causal particles to causal verbs, and higher complexity of verb phrases. In contrast, the Goal category was characterized by a higher use of causal verbs and a more formal, logical, and hierarchical thinking processes. This implies that Goal statements were generally expressed using causal, yet simpler linguistic structures (active language, simple causal statements), whereas the Motive category was characterized by a more complex language (i.e., more complex verb phrases, more passive expressions, more causal particles). Finally, we also see a unique profile of non-reflective statements (the Other category), which were characterized by a higher focus on personal topics and less driven by the perceptual processes. We also see a more frequent use of punctuation, which is likely caused by the use of emoticons in the non-reflective messages. Interestingly, on a linguistic level, we found a higher use of nouns, which requires a further study that we will conduct in our future work.

7 CONCLUSIONS

The contributions of this paper are twofold. First, we developed a classification system for categorization of students’ reflections in accordance with the coding scheme by Hulsmans et al. [35] which provides a moderate accuracy (accuracy of 89% and Cohen’s k of .51) over the chance level. The use of LIWC and Coh-Metrix features shows a great potential for understanding students’ reflective writings, which are based on well-established linguistic metrics of different psychological processes. Second, our study provides a detailed evaluation of the linguistic indicators of the different types of student reflection. Interestingly, the most significant predictor was the use of perceptual words (e.g., seen, view, saw) and the complexity of causal expressions (i.e., the ratio of causal particles to the causal verbs). We also found that basic N-gram features provided less value than highly theorized linguistic metrics from LIWC and Coh-Metrics analysis tools. Finally, our results also showed some benefits of utilizing the reflection context, which was in our case captured by a single variable that indicated the relative position of the statement in a sentence. As such, in our future work, we will focus on providing more a detailed operationalization of the annotation context. We will also examine the use of the system for provision of the real-time feedback to students, which is one of the most promising uses of learning analytics [28].

REFERENCES
