Using Lexical Analysis Software to Assess Student Writing in Statistics

1. INTRODUCTION

For over twenty years, there have been calls for improving Science, Technology, Engineering and Mathematics (**STEM**) education in the United States (American Association for the Advancement of Science, 2009; Committee on Undergraduate Science Education, 1999; Gess-Newsome, Johnston, et al., 2003; Kardash & Wallace, 2001; National Science Foundation, 1996; Ruiz-Primo, Shavelson, Hamilton, et al., 2002; Seymour, 2002; Seymour & Hewitt, 1997; Tobias, 1990). These calls have been paralleled in the statistics education community (Gal & Garfield, 1997). A common recommendation is to move STEM instruction away from teaching and assessing facts to helping students acquire deeper conceptual understanding and transferable problem-solving skills. Similarly, in 1992, the ASA/MAA Joint Curriculum Committee gave three recommendations for changing statistics courses, one of which was to incorporate more data and concepts and fewer recipes and derivations (Cobb, 1992). Furthermore, one of the recommendations of the Guidelines for Assessment and Instruction in Statistics Education (**GAISE**) college report is to stress conceptual understanding, rather than mere knowledge of procedures (Aliaga, Cobb, Cuff, et al., 2005).

Meaningful assessments that reveal student thinking are vital to these efforts (Pellegrino, Chudowsky, and Glaser, 2001) as emphasized by a second recommendation of GAISE: use assessments to improve and evaluate student learning (Aliaga, et al., 2005). Assessing conceptual understanding, however, is recognized as a "challenge faced by all educators in statistics education" (Gal & Garfield, 1997). To this end, there are efforts in science education devoted to developing concept inventories for formative assessment of students' understanding of important 'big ideas' in science (D'Avanzo, 2008; Knight, 2010; Libarkin, 2008). A similar undertaking in statistics is the Comprehensive Assessment of Outcomes of a First Statistics Course (**CAOS**). Concept inventories, like the CAOS, are typically multiple-choice assessments in which the distracters were derived from common student misconceptions. These misconceptions were generated by educational research on student thinking and alternative conceptions about the big ideas in STEM disciplines (Duit, 2009). These findings were obtained by asking students to construct explanations to questions either in interviews or writing.

The utility of multiple-choice concept inventories is that they are efficient to administer to large groups of students. Constructed-response questions, also known as open-response or short answer questions, in which students must write an answer in their own words, have been shown to reveal students' understanding better than multiple-choice questions (Bennett & Ward, 1993; Birenbaum & Tatsouka, 1987; Bridgeman, 1992; Kuechler & Simkin, 2010). For example, Nehm and Schonfeld (2008) showed that constructed-response scores have greater correspondence with clinical interview scores than multiple-choice test scores in the subject of natural selection in biology. The drawback to the use of constructed-response tests is the time and expertise needed to score them, particularly in large enrollment courses or large research projects. Recent advances in technology and natural language processing, however, have made

computerized analysis of writing possible and may facilitate the analysis of large numbers of written responses. In fact, the correspondence between clinical interview and constructed-response scores persists when the constructed responses are scored by computers, which are capable of capturing even students' "naïve ideas as accurately as human-scored measures" (Beggrow, Ha, Nehm, et al., 2013, pg. 14).

The Automated Analysis of Constructed Responses (**AACR**; http://www.msu.edu/~aacr) research group, consisting of researchers from seven universities with backgrounds in various STEM disciplines, has as a goal the creation of open response questions that can be scored using software to help us gain greater insight into student thinking about 'big ideas', such as evolution, energy, and genetics. To date, the work of the statistics subgroup of AACR has been to inform biology education researchers of the problems students have in understanding the meaning of the word *random* (Kaplan, Rogness, and Fisher, 2014). These misuses of the word *random* have been noted in the biology education literature by Garvin-Doxas and Klymkowsky (2008), who claim that genetic drift, a random process underlying evolutionary change in biology, understood by both students and working scientists and in the journal, *Nature*, in which Ochs (1990) was able to cite three papers published in the previous year that used the word *random* incorrectly.

In the next section of the paper, we describe the use of two different software packages, LightSIDE and Text Analysis for Surveys (**TAS**), for analyzing student open responses. Section 3 describes the methodology associated with the data collection and analysis of open-response data. These data were collected from undergraduate students' writing about the word *random* at the conclusion of an introductory statistics course. The analysis and results produced by the two packages will be contrasted with each other and with the results obtained from hand coding of the same data sets. The article will conclude with a discussion of the advantages and limitations of the analysis options for statistics education researchers and directions for future research using the software.

2. DESCRIPTION OF THE SOFTWARE

2.1 Light Summarization Integrated Development Environment (LightSIDE)

LightSIDE is a free and downloadable (http://www.cs.cmu.edu/~emayfiel/side.html) machine-learning software package developed at Carnegie Mellon University (Mayfield & Rosé, 2010). The development of the LightSIDE software was based on the computer science research field of machine learning, in which researchers investigate how to teach computers to mine texts and build models based on human generated training sets (for details, see Abu-Mostafa, 2012). Generally, the machine learning software is taught by the patterns found in the human scored training sets. The algorithm that is generated by the machine learning software from a training set is used to predict the categorization decisions of texts that have not yet been scored manually.

In practice, LightSIDE takes a set of human-scored text-based responses (for instance, a spreadsheet of responses that have been scored for the presence or absence of specified concepts) and discovers word patterns that account for human-generated scores. LightSIDE performs much of the difficult work of figuring out what elements differentiate an accurate response from an inaccurate response, or a response in which a series of words that represents a concept is present

or absent. LightSIDE next applies the rules it learned from human scoring on a new set of responses and determines how well the rules work using Kappa agreement values. This software enables users to extract features (e.g. words in texts) from text and to build the algorithm on how the features predict human decision (e.g. human scoring).

Using the LightSIDE program consists of three operations: 1) extracting features, 2) building models, and 3) predicting labels. In the first step LightSIDE mines important words from students' essays. The researchers are able to control machine learning parameters such as *n*-gram selection (e.g., unigrams, individual words, or bigrams, two word phrases, see Wang, Chang and Li, 2008), stemming (e.g., *predict* for *prediction*, *predictable*, and *predicted*), and removing stopwords (e.g. *a*, *an*, *so*, or *and*). LightSIDE detects *n*-grams that indicate both whether a response belongs or should be excluded from the category and assigns each of the detected *n*-grams a value indicating its importance as part of the inclusion or exclusion model.

In the second step of the analysis, building models, LightSIDE builds an optimal algorithm to predict the human graders' scoring of the responses into categories. LightSIDE is equipped with many types of support vector machines (**SVM**) (e.g., Sequential Minimal Optimization (**SMO**) or Library for Support Vector Machines (**LIBSVM**)) with which to build a scoring model. In this step of the analysis the researcher must choose a support vector machine from which LightSIDE will build its scoring model. Initially, the machine-learning experts who developed LightSIDE recommended the use of SMO (Moharreri, Ha, and Nehm, 2014). They claimed it was state-of-the-art because, as a revised version of the original SVM, it improved on the SVM process. In our experience, the choice of SVM used in model-building in LightSIDE should be experience-driven not rationale-driven; in fact, many researchers and projects using LightSIDE incorporate other algorithms (e.g. naïve Bayesian classifier) as part of the analysis (Mayfield, Adamson, and Rosé, 2013). In the initial analysis of the data presented in this paper, SMO showed the best performance in terms of providing the most reliable models for the data so it was used for this study.

In the third step of the analysis, predicting labels, researchers can use the model that LightSIDE built in step two to label new text (e.g. students' new responses to be scored). The results of LightSIDE's prediction are then downloadable into an Excel file. Previous results have shown inter-rater reliability between a human scorer and LightSIDE to be better than that between the human and a second independent human scorer (Beggrow, et al., 2013).

One advantage of using LightSIDE is that machine learning can extract text features and build predictive scoring models automatically, without the human labor that can be a highly time- and cost-consuming process (Nehm, Ha, and Mayfield, 2012). In addition, bigram (i.e. the combination of two words) and trigram (i.e., the combination of three words) functionalities are able to differentiate the meaning of two sentences that consist of the same words. Consider, for example, student descriptions of a skewed histogram. Students may describe the histogram as having *right skew* or *left skew*, with only one of the descriptions being correct. If the program parsed the writing into unigrams only, the word *skew* would be indicative of both a correct and incorrect response. The use of bigrams allows the program to read the bigram *right skew* as different from *left skew*, giving it the ability to characterize one phrase as correct and the other as

incorrect. A final advantage of LightSIDE is that it is released under General Public License (**GNU**). Therefore, it is free for everyone and modifiable by anyone.

One disadvantage of LightSIDE as a machine learning software, however, is that it requires data that have been hand-scored by humans because the software needs to learn from patterns in human decisions. Researchers need to collect and score an amount of data sufficient to build a reliable scoring algorithm. Sufficiency, in terms of the amount of data needed, depends on the complexity of the scoring algorithm, such as the diversity of features associated with a category. Moreover, the larger the amount of human-scored data that can be provided, the better the software is enabled to learn the patterns more effectively and accurately (see Ha, Nehm, Urban-Lurain, et al., 2011). Researchers need to investigate empirically how much data LightSIDE needs to build an accurate and reliable model. Another disadvantage of LightSIDE is that it is difficult for users to intentionally control and modify the algorithm that the software generates. For example, an algorithm typically uses all features extracted from text to build the predictive scoring model when a subset of the extracted features would be sufficient to create a model with similar reliability. Although LightSIDE consists of several functions to select features (e.g., removing noisy features), it is still difficult to manually control the features that interact to build the algorithm because the software typically extracts and uses a large number of features from text.

2.2 IBM/SPSS(R) Text Analytics for Surveys - TAS

IBM SPSS Text Analytics for Surveys (TAS) is lexical analysis software that was originally designed for processing survey responses from marketing research (i.e. responses to questions such as, "What can be improved about product Y?"). There is current work that extends the utility of this software to analyze student responses in STEM education research (Ha & Nehm, 2011; Haudek, Kaplan, Knight, et al., 2011; Haudek, Prevost, Moscarella, et al., 2012; Nehm & Haertig, 2012). For this report, we have used IBM SPSS Text Analytics for Surveys, v. 4.0 (SPSS, 2010), which is commercially available at a cost of 2,400 USD (at the time of writing).

TAS uses linguistic-based extraction to identify terms and/or phrases from blocks of machine-readable text. These terms may be single words (e.g. *random*) or they may be phrases (e.g. *random sample* or *random sample of students*). As a group, terms and phrases are referred to as lexical tokens or, simply, tokens. The software recognizes tokens by using built-in and/or customizable libraries, which can be thought of as dictionaries. Therefore, to investigate student writing in STEM, some work must be done to build a custom library that contains unique tokens in the student responses. For example, the phrase *equal chance* had to be added to the library used by the current project. These custom libraries are re-usable, however, and shareable between different text analysis projects; so custom libraries already built for one project need less revision for their subsequent use in other projects.

Once the tokens are extracted, similar terms and phrases can be grouped together in categories. Categories may also contain functions, which combine tokens via Boolean operators. In a biology education setting, researchers created a category for "Random" defined as a response containing the word *random* **or** the combination of the words *by* **and** *chance*. Thus the category definition can be written symbolically as {random | (by & chance)}. Category grain-size in TAS

can be refined by the user, resulting in categories that may be very detailed (possibly containing only a single term) to extremely broad (possibly aligning with a scoring rubric level).

TAS also contains linguistic and frequency-based algorithms to generate categories automatically (SPSS, 2010). These categories, however, do not always align well with the desired grain size or purpose of the research question, so some effort must be made to verify and hone categories by subject-matter experts. Therefore, categories are usually developed iteratively through careful examination of student writing and the lexical analysis output. TAS supports this iterative refinement, but ultimately an expert must make the decisions about the categories. The output from TAS is a series of binary variables based on the generated lexical category, whether a given response is present or absent in each category. These variables can be exported in a manner suitable for further statistical analysis or modeling techniques. Similar to findings using LightSIDE, previous research using TAS has shown that results from TAS can be used to create predictive scoring models on par with or better than inter-rater reliability measures between two independent human coders (Ha & Nehm, 2011; Haudek, et al., 2012; Nehm & Haertig, 2012).

One advantage of using the TAS software is that it supports the Grounded Theory method of qualitative research (see, for example, Corbin & Strauss, 2008). The researcher must, at some level, become immersed in the student writing, insomuch as the researcher must identify terms in the writing in order to build libraries, and make decisions about how to group these terms into categories. These tasks necessitate the reading of student writing. In this way, the researcher may notice novel and/or emergent ideas in students writing. This approach to research requires little a priori knowledge of the kinds of ideas students are likely include in their explanations.

Another advantage of using TAS is the ability to modify category grain-size. Researchers may have different purposes for the analysis of responses to different questions. Our experience has been that categories that represent one homogenous idea give the best analytic results. Depending on the research question, however, the size and scope of one idea can change drastically. TAS gives the researcher the ability to categorize the same set of terms differently by simply adjusting the categories. For example, as will be discussed later, students define *random* as something that is *independent, representative* or *without bias*. In our original coding, these ideas were grouped into the same category. TAS allows the user to easily separate or combine these three ideas, thereby allowing refinement of categories during the course of the project.

The last advantage of using TAS is the ability to create diagrams of the lexical categories, called webmaps, from within the software. It should be noted that there are other software programs that can generate webmaps using the analytic outputs of either TAS or LightSIDE. This, however, requires multiple exporting and formatting steps. We have found utility in having the webmap feature available within the TAS lexical analysis software itself. Webmaps are a depiction of the co-variance of all or a selected sub-set of categories. Webmaps use nodes to represent the lexical categories and lines connecting nodes to represent responses shared between the two categories (for an example and further discussion see Figure 2 and Section 5.3). Generating webmaps allow a user to see immediately the connection (or lack thereof) between any two or more categories. This depiction is useful during the iterative process of category refinement.

One disadvantage to using TAS is the amount of time in human labor needed to get refined output from text analytics (further discussion in Nehm & Haertig, 2012). As described previously, the software user has two main tasks using TAS: custom library creation and category refinement. The exact amount of time needed for library creation is dependent on the types of words one wishes to extract from responses. If the desired words are common terms, little effort is needed in library creation. On the other hand, if the desired words are scientific jargon, more effort is necessary, especially if there has been no effort to customize a library in that domain previously. As it relates to category refinement, the software user must confirm that each category represents only a single homogenous idea, at the desired and appropriate grain-size for the project. It has been our experience that time used for library building decreases as custom dictionaries are re-used and revised. As such, most of the human involvement in text analysis projects using TAS is applied in the creation and iterative refinement of categories.

Another disadvantage of using TAS is the monetary cost for licensing the software. It should be noted that other software products (e.g. IBM SPSS Modeler; SPSS, 2011) are available for a lower cost and have similar text analytics features. Modeler also allows the connection of text analysis results with a variety of computerized model building techniques in a single software environment.

3. METHODOLOGY

3.1 Description of the Data

The data that will be used to illustrate the two software packages consist of sentences and definitions written by undergraduate students at the end of a one-semester introductory statistics class. In particular, the students were given a questionnaire on which appeared the following set of questions:

a. Write a sentence with the word "random" using its primary meaning to you in statistics.
b. Provide a definition for the word "random" using its primary meaning to you in statistics (i.e. maintaining the same meaning as you used in the prior sentence).

The data discussed in this paper were collected as part of a large-scale study conducted during the fall semester of 2008 at three universities in the United States one located in the Southeast and two in the Midwest (for more details about the data collection and study design see Kaplan, Fisher, and Rogness, 2010 or Kaplan, Rogness, and Fisher, 2014). Two of the institutions are classified as having high research activity, one of which had large enrollment of over 40,000 undergraduates and the other with 16,000. The other institution is a medium-sized comprehensive university. At the largest institution, introduction to statistics courses are taught in lecture format. For three hours each week, the students meet in lecture halls with approximately 120 students per lecture. The students attend an additional hour of recitation with a graduate teaching assistant once per week in classes of 30 students. At the other two institutions, enrollment in the classes is 30 – 40 students per classroom and at one of those institutions the class met at least one hour per week in a computer lab. At all institutions the course in which the data were collected is a service course for students in a variety of majors including nursing and the social sciences. The topics covered include descriptive statistics,

confidence intervals, hypothesis testing, introduction to correlation and regression, and Chi Square Test of Independence.

The total number of subjects for the large-scale study was 859, with 14 different instructors across the three institutions. Of these subjects, 534 gave a sentence and definition for *random* in the statistical sense in answer to Questions a and b. These 534 responses are called the "Complete data set" in this paper. A simple random sample of all of the responses from the Complete data set was selected for the initial phase of data analysis. This sample of responses is called the "Subset sample" and contained 65 responses about the statistical meaning of *random*. A second sample of responses was collected from the class taught by the first author in the fall 2008 semester. These 82 responses were added to the data corpus; this sample is called the "One-Class sample."

3.2 Analysis Procedures

The research team used three different methods to code student responses: hand coding, LightSIDE software and TAS software. Results of hand-coding were used as variables in scoring models using LightSIDE and helped inform computerized categorization using TAS. Procedures for each of the three coding methods are described in this section.

3.2.1 Manual Coding of the Data

The research team used data from a pilot study to create coding categories for the student sentences and definitions for the word *random*. One researcher read all the responses and used the responses to create categories. Once the first researcher had finished creating coding categories for the definitions and had coded all the responses, draft versions of coding categories and the student responses were then sent to one other researcher. That researcher independently coded the responses and suggested modifications and edits to the coding categories. The three members of the research team discussed the responses on which the two independent coders disagreed and modified the coding rubric as necessary. After this discussion there was 100% agreement between the three researchers.

The coding rubric for student statistical uses and definitions of *random* contains six categories: 1. "Uncoded," 2. "By Chance," 3. "Without Order or Reason," 4. "Unexpected, Unpredictable," 5. "Without Bias, Representative," and 6. "Equally Likely." The categories with higher assigned numbers are closer to a statistically sound understanding of the word *random* and each response was coded into only one category, corresponding to the category that was the highest that could be justified by the coder. It may seem peculiar that the most advanced coding category in the rubric represents a common misconception about random processes, that the outcomes are equally likely (Fielding-Wells, 2014). At the time, however, this category was the only one in which responses mentioned likelihood or anything closely related to probability, and there were no responses that mentioned probability without also stating the condition of equal likelihood. The research team then used the Subset sample ($n = 65$) to validate the rubrics created with the pilot study data. Each of the 65 definition and sentence pairs was independently coded by two researchers, with an initial agreement of 72%. All disagreements were discussed by the three researchers and the coding rubric was amended as necessary until there was 100% agreement on

all of the instruments. The One-Class sample was coded by the first author using the coding rubric developed from the pilot study and Subset sample coding (for more detail about the categories and hand coding see Kaplan, Fisher, and Rogness, 2010 and Kaplan, Rogness, and Fisher, 2014).

3.2.2 Coding of the Data Using LightSIDE

To create training sets of data for LightSIDE, each response in the subset and One-Class samples was coded independently by three researchers as having or not having (presence or absence of) elements corresponding to each of the five categories used in the original hand scoring: "By Chance," "Without Order or Reason," "Unexpected, Unpredictable," "Without Bias, Representative," and "Equally Likely." All disagreements were discussed by the three researchers until 100% agreement was reached. The responses from the One-Class sample were then analyzed via LightSIDE. Features were extracted for each of the five categories using the settings: unigrams, line length, remove stopwords, stem and treating features as binary. Models for each category were built using the SMO SVM and validated using a 10-fold cross validation process.

This work relied on the use of the Cohen's kappa statistic to quantify the correspondence between human and computer scores (Bejar, 1991). Cohen's kappa values are widely used to measure inter-rater agreement. Considering the computer as a rater, the inter-rater agreement reliability captured by Cohen's kappa is an appropriate method for human and computer correspondence measures. Calculated Cohen's kappa coefficients can range from -1.0 to 1.0. A kappa value of 0 indicates no agreement between two raters (and negative values indicate that the observed agreement is lower than would be expected by chance). Literature has suggested several possible cutoff values of Cohen's kappa and how to interpret Cohen's kappa; in particular we used Landis and Koch's (1977) and Fleiss (1971)'s suggestions. Landis and Koch (1977) suggested that kappa values between 0.41 and 0.60 were considered moderate, values between 0.61 and 0.80 were considered substantial, and those between 0.81 and 1.00 were considered almost perfect. Fleiss's (1971) guidelines are more generous than Landis' and Koch's (1977) guidelines. Fleiss (1971) suggested that kappa values over 0.75 were considered as excellent, scores between 0.60 and 0.75 were considered good, values between 0.40 and 0.59 were considered fair, and values less than 0.40 were considered poor. Therefore, kappa values of 0.81 or greater were the target, but kappa values greater than 0.75 were considered to be the benchmark for kappa values in this study.

When a model built in LightSIDE, based on the Subset sample using the initial settings described previously, did not reach a kappa level of at least 0.75, several settings were changed in an attempt to create a model that would function better. One setting that addressed the small size of the data set was to set the threshold to three, rather than five. This encourages LightSIDE to recognize words that appear in only three of the responses in a given category, rather than requiring five responses to contain the word. Another setting that was modified was to specify that LightSIDE consider bigrams, or two-word phrases. For example, in the category for random sample, this allowed the software to recognize the phrase *random sample*, rather than each of the words, *random* and *sample*, individually. When these modifications still did not yield a model that reached a kappa level of 0.75, the two data sets, One-Class and Subset, were merged to

create a larger training set. While this tweaking of the tuning parameters might result in a model that overfits the data, we were able to test the models built only on the One-Class sample for overfit using the Subset sample. Unfortunately, the models built on the combined One-Class and Subset samples could not be tested for overfit due to the lack of an additional hand-coded data set. The details of the models along with the results of the coding of the Complete data set are discussed in detail in Section 4.2.

3.2.3 Coding of the Data Using SPSS-TAS

To begin the coding of the data using SPSS-TAS, the One-Class sample data were read into the program, which extracted terms based on the libraries built into the software. After reviewing the terms that had been extracted, the first author created a new library, called the Random Library, for the terms that the software had not extracted. This process was informed by the previous manual coding and use of LightSIDE for coding. Some of the terms that were added were agents of randomization, such as *spinner, dice*, *hat* and *coin.* Other additions included actions, such as *flip a coin* or *choose out of a box*; the built-in SPSS-TAS libraries do not contain verbs or actions due to their construction for use in analyzing marketing surveys. Another class of additions to the library was phrases having to do with likelihood, such as *equally likely, same chance*, or *equal opportunity.*

Once the most obvious additions to the library had been made, categories were built for each of the five categories that had been previously identified in the data: "By Chance," "Without Order or Reason," "Unexpected, Unpredictable," "Without Bias, Representative," and "Equally Likely." In addition, categories were built for the phrase *random sample* and for agents of randomness as described previously. All categories were built using rules. For example, the category "Equally Likely" contained two rules: The first placed all responses that used the word *probability* into the category and the second placed all responses that used the words *equal, equally,* or *same* as a modifier of one of the words *chance, opportunity,* or *likely.* After the seven categories had been created and the responses were categorized, the categorizations were checked against the manual coding and the results of the coding by LightSIDE using Cohen's kappa values as specified in the previous section. Terms were added to the Random Library and the rules for the categories were modified in an iterative process until the coding of the One-Class sample by SPSS-TAS produced kappa values of at least 0.75.

The Random Library and categories were saved in a way that allowed the researchers to read in and analyze the Subset sample data using the same library and categories. Tokens were extracted and responses were categorized based on the built-in and Random libraries and rules that had been created using the One-Class sample. These categorizations were compared to those that had been done using manual coding and a second iterative process was used to augment the Random Library and rules so that the results of the SPSS-TAS coding of the Subset sample produced sufficiently high kappa values when compared to the coding that had been done manually. The iterative process continued until the kappa values for all seven categories for both of the data sets was sufficiently high. Once the library and rules were created based on the smaller training sets, the Complete data set was read into TAS and analyzed using the library and rules. These results are discussed in detail in Section 4.3.

3.3 Comparing Results from LightSIDE and TAS

Once the models created in LightSIDE and TAS reached acceptable kappa levels with human hand coding for the Subset and One-Class samples, the models were applied to the Complete data set. Using the results of the hand coding, the number of expected responses in each category were calculated so the number of responses indicated by each of the two programs could be compared to the expected number. In addition, inter-rater reliability and percent agreement were calculated for the coding in LightSIDE when compared to TAS. These results are presented in Section 4.4.

4. RESULTS

4.1 Results of Hand Coding

Table 1 provides the hand coding results of the Subset and One-Class samples and examples of student generated sentences and definitions for the word *random* from which the categories were derived. Recall that each response was placed into exactly one category, corresponding to the highest possible category into which the coder determined the response belonged. The results, therefore, reflect disjoint categories without retaining information about connections between the categories (for more details about the categories, see Kaplan, Fisher, and Rogness, 2010 and Kaplan, Rogness, and Fisher, 2014).

In order to create and test categorization models for the data in the two computer programs, the data had to be re-coded by hand indicating every category into which each response could be classified. During the analysis of the data using TAS (described later), two additional categories were identified: “Random Sample” and “Agents.” These categories were included in the recoding of the data. In addition, when the data were recoded into multiple categories, the research team stopped accounting for any responses that were not coded into any category. The results of the hand coding are presented in Table 2; the same results are presented graphically in Figure 1. It is still apparent that the subjects in the One-Class sample tended to mention more statistical aspects of randomness, such as equal likelihood, unbiased, and agents than did the subjects in the Subset sample. In contrast, subjects in the Subset sample were more likely to mention random sampling and elements of without order, reason or pattern than the One-Class sample. This analysis, however, still does not provide insight into the relationships between the categories and, due to the prohibitive amount of time such hand coding would have required, the Complete data set was never hand coded.

Table 1: Sentences and definitions for random given by students

Definition Category	Subset Sample (n = 65)	One-Class Sample (n = 82)	Example
Uncoded	12%	6%	**Sentence:** We used a random variable today. **Definition:** random: unknown
By Chance	4%	9%	**Sentence:** For the survey, a random sample was picked. **Definition:** by chance that something occurred.
Without Order or Reason	39%	12%	**Sentence:** It was a random sample, which provides independence. **Definition:** Random: persons were chosen not based on any reason.
Unexpected, Unpredictable	14%	9%	**Sentence:** I was picked for a random sample. **Definition:** Not pre-determined.
Without Bias, Representative	23%	24%	**Sentence:** The sample population is a random sample. **Definition:** Sample is equally representative of all groups of the population.
Equally Likely	8%	40%	**Sentence:** We took a random sample of the students. **Definition:** everyone was equally likely to be chosen for the sample.

Table 2: Hand coding of the data (multiple categorizations allowed)

	Percent in Sample	
Category	Subset Sample (n = 65)	One-Class Sample (n = 82)
By Chance	9.2%	20.7%
Without Order or Reason	21.5%	13.4%
Unexpected, Unpredictable	12.3%	8.5%
Without Bias, Representative	35.4%	31.7%
Equally Likely	9.2%	36.6%
Random Sample	56.9%	41.5%
Agents	1.5%	30.5%

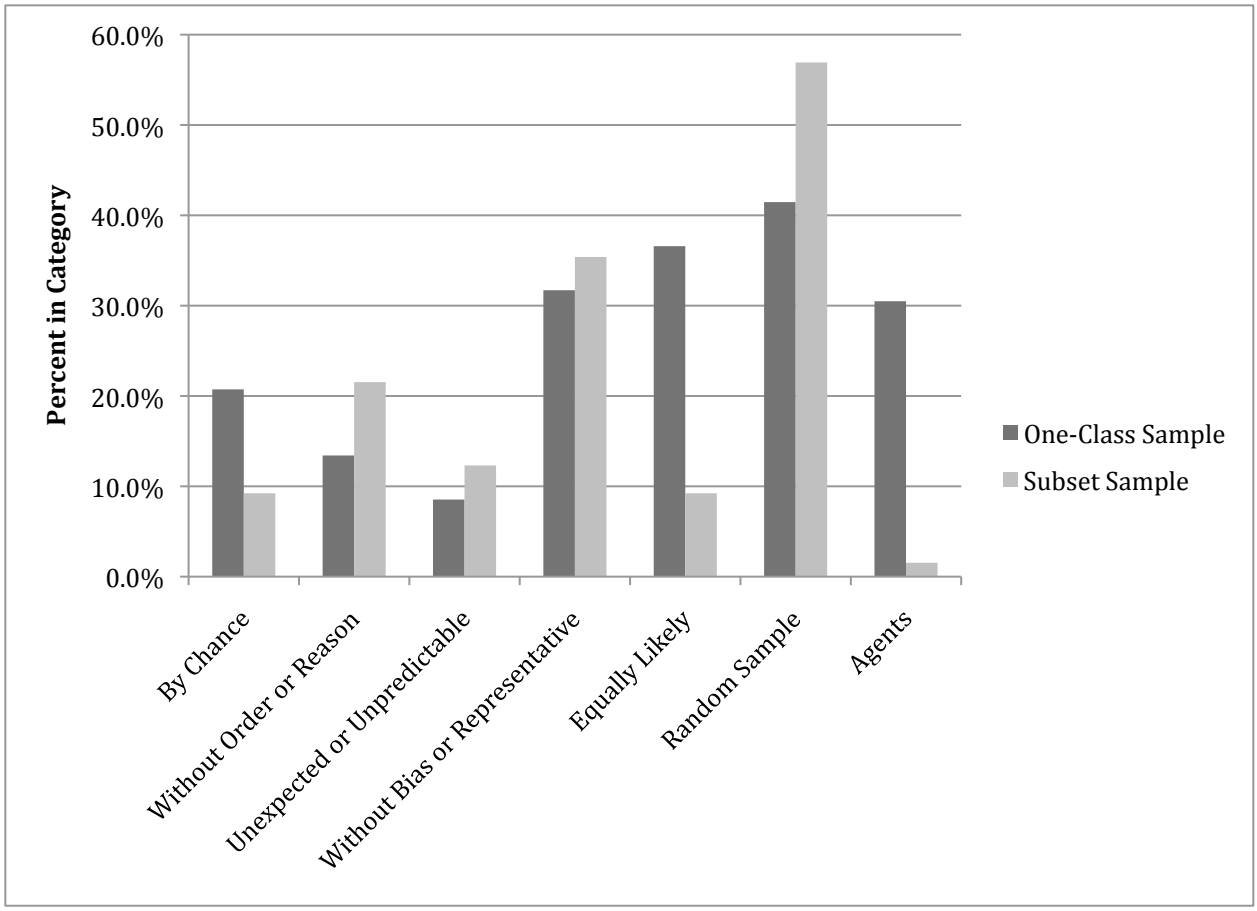

Figure 1: Results of hand coding of the data (multiple categorizations allowed)

4.2 Results of Coding using LightSIDE

Table 3 provides the results of the initial analysis of the One-Class sample data through the first two steps in LightSIDE: extracting terms and building models. Notice that the models built for 6 of the 7 categories (all except "Without Bias, Representative") correctly categorized over 85% of the responses. Unfortunately, only the models for the 3 categories "By Chance," "Equally Likely," and "Random Sample" had kappa values meeting the stated criteria. The low kappa values for the other categories are the result of the high number of false negatives; in other words, the software has missed one-third to one-half of the responses that should have appeared in the category.

When the model for the category "By Chance" was applied to the Subset sample data, every response was correctly categorized (i.e. 100% correct, kappa = 1.0); this model was therefore considered to be functioning properly and was applied to the Complete data set. The other two models, "Random Sample" and "Equally Likely," did not fare as well when applied to the Subset sample, classifying 50.8% and 73.8% of the responses correctly with kappa values of 0.0989 and -0.5476, respectively. In fact, the model for "Equally Likely" was not able to detect any of the 6 responses that should have been in the category. The problems with the "Random Sample" model were resolved by having LightSIDE consider bigrams, or two-word phrases, when extracting features from the data. This new model correctly categorized 92.7% (76) and 95.4%

(62) of the responses in the One-Class and Subset samples, respectively, with corresponding kappa values of 0.8493 and 0.909 (see Table 4).

Table 3: Results of the first modeling attempt in LightSIDE using the One-Class sample

Category	Kappa	Correctly Classified	Number in Category	False Positives	False Negatives
By Chance	0.9258	80 (97.6%)	17	1	1
Without Order or Reason	0.5847	75 (91.5%)	11	2	5
Unexpected, Unpredictable	0.4617	76 (92.7%)	7	2	4
Without Bias, Representative	0.4250	62 (75.6%)	26	9	11
Equally Likely	0.8949	78 (95.1%)	30	2	2
Random Sample	0.8264	75 (91.5%)	34	5	2
Agents	0.6385	70 (85.4%)	25	4	8

The problems with the "Equally Likely" model were solved by setting the extraction threshold in LightSIDE to three instead of five. In other words, only three responses in the category needed to contain a word in order for that word to be extracted as informative of the category. This new model correctly categorized 97.6% (80) and 98.5% (64) of the responses in the One-Class and Subset samples, respectively, with corresponding kappa values of 0.9474 and 0.9008 (see Table 4). The threshold was also set to three for the category "Agents." This raised the percent of responses correctly classified in the One-Class sample to 92.7% (76) and provided an acceptable kappa value of 0.8234. It was noted, however, that the model had difficulty recognizing the words *computer* and *die* as representative of the category. Furthermore, the one response in the Subset sample that should have been in this category was not detected. Because of the relative lack of responses fitting this category in the Subset sample, and the supposition that future data with more examples in this category would analyze better, no further work was done to improve the model.

When similar solutions were not effective for the three remaining categories, "No Reason or Order," "Unexpected, Unpredictable," and "Without Bias, Representative," all of the hand coded data, the One-Class sample and the Subset sample, were combined into one data set with 147 responses. For each category, the threshold in LightSIDE was set to three. The model generated for the category "No Reason of Order," with a kappa value of 0.8456, correctly classified 95.9% of the responses, generating only one false positive and missing five of the 25 responses that should have been included in the category. The model for "Unexpected, Unpredictable," with a kappa value of 0.7822, generated no false positives, but misclassified five of the 15 responses that should have appeared in the category. The software generated a model to predict the category "Without Bias, Representative" with a kappa value of 0.7859 that had a misclassification rate of 9.5%, missing nine of the 51 responses that should have been included in the category and generating five false positives. While two of the models did not reach the highest standard for the cutoff of kappa values, all did exceed a kappa value of 0.75. In addition, the models had a tendency to produce a relatively large number of false negatives. That said, all

models correctly categorized over 90% of the responses in the two data sets. These models were, therefore, considered sufficient. Summary information about the models is given in Table 4.

Table 4: Results for the final models generated using LightSIDE

Category	*n* required	One-Class sample		Subset sample	
		Kappa	Correctly Categorized (*n* = 82)	Kappa	Correctly Categorized (*n* = 65)
By Chance	82	0.9258	97.6%	1	100%
Equally Likely	82	0.9474	97.6	0.9008	98.5%
Random Sample	82	0.8493	92.7%	0.9090	95.4%
Agents	82	0.8234	92.7%	0	98.5%
		Model Kappa		Correctly categorized	
Without Order or Reason	147	0.8456		95.9%	
Unexpected, Unpredictable	147	0.7822		96.6%	
Without Bias, Representative	147	0.7859		90.5%	

4.3 Results of Coding using TAS

As discussed in Section 3.2.3, the first step in using TAS is to read in the data and allow the software to extract lexical tokens based on the libraries built into the software. Once this was done, a new library was created using terms related to randomness. Next, rules corresponding to each of the seven coding categories were created. An iterative process was used to update the library and rules until the number of responses in each of the categories according to TAS was approximately equivalent to the number of responses when categorized by hand. The process included not only adding terms to the library, but also adding synonyms, such as words invented by students like *biasedly* and *biasly* for *bias* and actual words like *pre-determine* and *undetermined* for *predetermined*.

Table 5 shows the rules that were used in TAS to classify the One-Class and Subset samples. The rules are created using standard logical operations (& for *and*; | for *or*) as well as standard use of parentheses and order of logical operations. The asterisk (*) indicates that a part of a word or phrase should be searched for and responses including any word or phrase beginning with the subpart should be included in the category. When a phrase, such as *by chance*, appears in a rule with no operator between the two words, this indicates that TAS extracted the full term as a bigram. If an operator appears between two words in a rule, such as *random & sample*, this allows the software to correctly classify responses when each term is extracted individually, rather than as the bigram *random sample*.

Two of the categories are described by only one rule: “Agents” and “By Chance.” The other categories contain multiple rules. While the *or* operator can be used to combine rules, rules that contained different aspects of the category were kept separate so they can be analyzed individually at a later date. For example, in the category “Without Bias or Representative,” the

first rule contains the aspects of the category related to representativeness, the second rule contains aspects of independence or no relationship and the last three rules contain various elements associated with the concept of bias.

Table 5: Final set of rules used in TAS classification

Category	Rules
By Chance	by chance | equal chance | same chance |(fair & chance) | fair chance
Without Order or Reason	1. ((without | no) & (order | reason | pattern | organization)) | unplanned 2. no order | no reason | no pattern 3. reason | order | pattern
Unexpected or Unpredictable	1. Unpredicted 2. Foresee 3. not predictable | predict | (no & predetermined)
Without Bias or Representative	1. representative | representative of all groups | equally representative 2. independen* | (no* & (relation | influence | association)) | any relation | no relationship 3. [human motive + <Contextual>] | human & impact | ((without | no*) & preference | influence*) 4. (no & bias) | ([bias + <Contextual>]) | ([bias + <PositiveBudget>]) | (without & bias) | unbiased | biased | not biased | non-biased | skew 5. Judgments of bias
Equally Likely	1. equal chance | equal opportunity | equal probability | same chance | same probability | equal likely | equal likely chance | (equal & shot) 2. probability
Random Sample	1. random sample of * | random sampling of * 2. random & sample 3. random sample 4. random sampling
Agents	coin | coin flip | hat | hat to get a random sample | hat to make a random sample | names from the hat | computer | calculator | die | dice | cards

Table 6 provides the summary statistics for the TAS models when compared to the hand scoring of the data. Notice that almost all of the models correctly classify at least 95% of the responses, with kappa values near 0.85 or higher and that all of the models have kappa values higher than the lower threshold of 0.75. The model that fares the worst is the model for "Equally Likely" when applied to the Subset sample. The reason for the low kappa paired with the relatively high percent of correct classifications (96.9%) is because the only two misclassifications were false negatives. There were only six responses in the data set that had been hand coded into the category "Equally Likely," and TAS was only able to find four of them, missing two or one-third of the responses that should have been found. When the two responses were reread and reconsidered, there was no clear modification to the rules or library that could be used to train TAS to correctly classify the two responses without creating the possibility of more false positives. Since the model had a sufficiently high kappa value and extremely high correct classification rate, the seven models were considered adequate and these rules and the

corresponding library were used on the Complete data set. These results will be discussed in Section 4.4.

Table 6: Kappa values for TAS models when compared to hand scoring

	Subset Sample (n = 65)		One-Class Sample (n = 82)	
Category	Kappa	Correctly Classified	Kappa	Correctly Classified
By Chance	1.0000	100%	0.9924	97.5%
Without Order or Reason	0.9040	96.9%	0.8483	96.3%
Unexpected, Unpredictable	0.9247	98.5%	0.9165	98.8%
Without Bias, Representative	0.9003	95.4%	0.8577	93.9%
Equally Likely	0.7841	96.9%	0.9217	96.3%
Random Sample	0.9688	98.5%	0.9502	97.6%
Agents	1.0000	100%	1.0000	100%

In addition to categorizing responses, the TAS software can create webmaps of the extracted categories. Examples of webmaps for the two data sets and all seven extracted categories are shown in Figures 2 and 3. The webmaps show not only the ideas and phrases present in the responses, but also the connections between the categories that tend to exist in the data. The webmaps include nodes for each category. The node sizes are based on the number of responses in that category with larger nodes indicating more responses. The largest nodes for the One-Class sample were for the categories “Random Sample” and “Equally Likely” and the largest nodes for the Subset sample were for “Random Sample” and “Without Bias, Representative.”

The thickness of the line that connects two nodes indicates the number of responses that include both of the connected categories. The webmaps show additional differences between the two samples with respect to the students’ writing about *random*. In the Subset sample webmap, (Figure 3) the strongest connections (denoted by the thickest lines) are between the three categories of “Random Sample,” “Without Bias or Representative,” and “No Reason or Order.” In contrast, the strongest connections on the One-Class sample webmap (Figure 2) include “Random Sample,” “Equal Chance,” and “Agents.” The thick connecting lines on the One-Class sample webmap include the statistical ideas underlying *random*, whereas the Subset sample webmap connections mirror a more colloquial use of *random.* Furthermore, the webmap for the One-Class sample shows many more connections between the ideas associated with randomness on the part of the subjects (for more detailed discussion of the differences between the two samples see Kaplan, Rogness and Fisher, 2014).

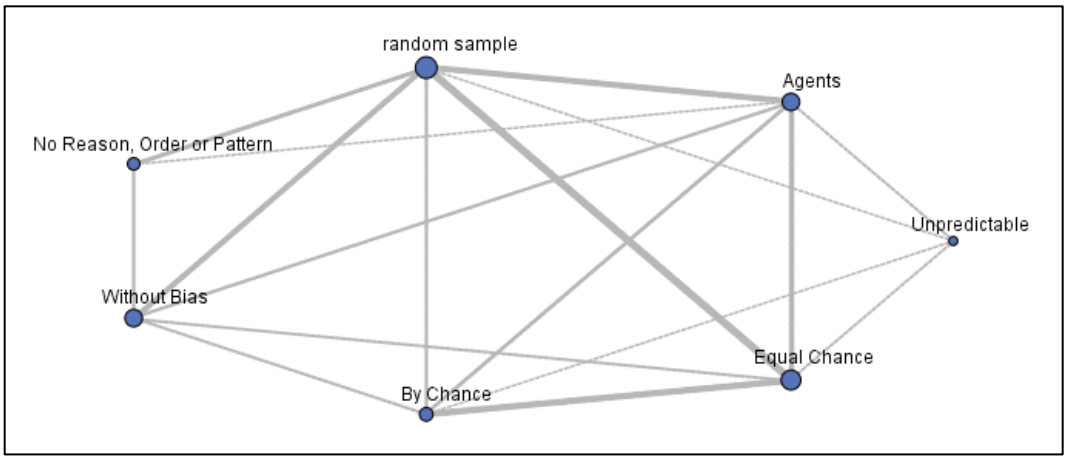

Figure 2: Webmap of categories for the One-Class sample

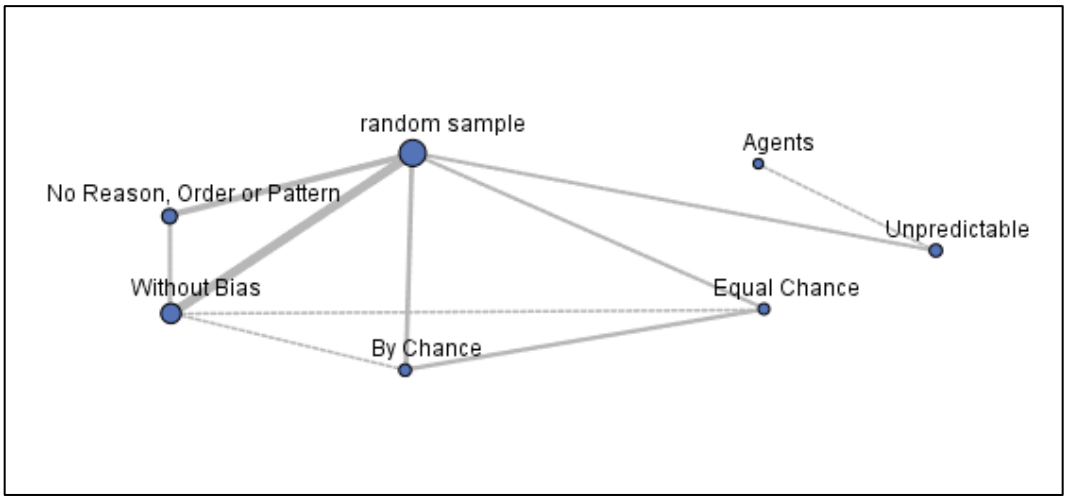

Figure 3: Webmap of categories for the Subset sample

4.4 Comparing Results of Coding using LightSIDE and TAS

In this section we compare the results of coding by the two computer programs. Tables 7 and 8 compare the coding of the two small data sets, Subset and One-Class respectively, by the software packages. Note that while both programs identified approximately the same number of responses in each category and approximately the same number as were indicated by hand coding, the kappa values and percent agreement indicate that the two programs did not select the same subset of responses in each category. That said, there are several issues apparent from the tables. First, both programs tend to under select responses in the "Unexpected, Unpredictable" category. This was true across both samples. While the kappa value representing inter-rater reliability between the two programs for this category was sufficiently high when applied to the Subset sample, it was not when applied to the One-Class sample. The opposite was true for the categories of "Agents" and "Without Bias, Representative," although the issue with the category "Agents" is an artifact of the fact that there is only one such response in the Subset sample and LightSIDE was unable to detect it.

Table 7: Comparison of coding by LightSIDE and TAS: Subset sample

	Number Identified (n = 65)				
Category	TAS	LightSIDE	Hand	Kappa (TAS v SIDE)	Percent Agreement
By Chance	6	6	6	1	100%
Without Order or Reason	12	10	14	0.8908	96.9%
Unexpected, Unpredictable	7	5	8	0.8168	96.9%
Without Bias, Representative	22	23	23	0.6262	83%
Equally Likely	4	5	6	0.8710	98.5%
Random Sample	36	36	37	0.9377	96.9%
Agents	1	0	1	0	98.5%

Table 8: Comparison of coding by LightSIDE and TAS: One-Class sample

	Number Identified (n = 82)				
Category	TAS	LightSIDE	Hand	Kappa (TAS v SIDE)	Percent Agreement
By Chance	15	17	17	0.8448	95.1%
Without Order or Reason	12	11	11	0.7472	93.9%
Unexpected, Unpredictable	3	5	7	0.4760	95.1%
Without Bias, Representative	25	24	26	0.9127	96.3%
Equally Likely	31	30	30	0.8695	93.9%
Random Sample	36	34	34	0.8505	92.7%
Agents	25	23	25	0.8234	92.7%

The models built in LightSIDE and TAS were then used to code the Complete data set containing 534 responses. The results comparing the categorization of the Complete data set are provided in Table 9. The Subset sample is a randomly selected subset of the Complete data set, so the percent of responses observed in the Subset sample for each category was used to predict the expected number of responses in the Complete data set. Notice that both programs under-predict when compared to the expected number of responses for all categories except "Agents," for which both models significantly over-predicted. Notice also that the most egregious of the under-prediction is in the category "Unexpected, Unpredictable," an outcome that might have been anticipated based on the results shown in Tables 7 and 8. Notice that the models for the three categories that were problematic in the coding of the smaller data sets remain problematic when applied to the larger data set. It should also be noted that two of the problematic models, "Without Bias, Representative" and "Unexpected, Unpredictable," were models for which LightSIDE required a larger data set to create. Furthermore, when coding the category "Without Bias, Representative" using TAS, it was apparent that this category is actually a merging of three ideas: bias, independence and representativeness. These issues may underlie the lack of inter-rater reliability between the computer programs.

Table 9: Comparison of coding by LightSIDE and TAS: Complete data set

	Number Identified (n = 534)				
Category	TAS	LightSIDE	Expected	Kappa	Percent Agreement
By Chance	32	40	49	0.8810	98.5%
Without Order or Reason	87	102	115	0.8267	94.9%
Unexpected, Unpredictable	9	28	66	0.4730	96.4%
Without Bias, Representative	122	167	189	0.6662	86.7%
Equally Likely	36	41	49	0.8181	97.5%
Random Sample	237	244	303	0.8903	94.6%
Agents	12	29	8	0.2191	94.2%

5. DISCUSSION

5.1 Summary of Findings

The results presented in Section 4 indicate that both software programs LightSIDE and TAS provide a means for analyzing open response data generated by statistics education research data. While not all of the models achieved the desired kappa level, we must consider the size of the One-Class and Subset samples. Previous research, such as that reported by Ha and Nehm (2011), used nearly 2500 hand coded responses to create computer scoring models as compared to the 82 or 147 used in this attempt. The models that functioned the best, those for the categories "By Chance," "Random Sample," "Agents," and "Equally Likely," were those that had fewer rules or elements associated with the categories. In addition, it was easier for the software to create models for the categories that appeared more often in the training set. In contrast, when we examined the categories for which the models did not function as well, "Without Bias, Representative," "No Reason or Order," and "Unexpected, Unpredictable," we found these to be multidimensional when compared to the other categories. That is, each of these categories contained several ideas. For example, "Without Bias, Representative" comprised responses of three distinct types: those containing the concept of bias, those containing the concept of independence and those containing the concept of representativeness. The authors are confident that hand coding of the Complete data set and/or splitting the problematic categories into their component parts, neither of which has been undertaken as of this time, would resolve the issues with the models for the more complicated categories.

5.2 Advantages and Limitations of the Coding Methods

Both software packages require a significant amount of hand scored data: LightSIDE to generate a training set so the software can detect categories and TAS to generate a data set with which to compare results and illuminate tokens and rules that need to be added to the software libraries

and categories. Thus, hand scoring is necessary. Hand scoring is also useful in that it helps the researcher to understand the structure of the data and the categories that may exist in the data. Three researchers spent a considerable amount of time generating the original rubric categories for the definitions of *random* from the student responses. That work was necessary to create a foundation from which to build models in LightSIDE and categories in TAS. That said, hand coding is time consuming, so a reasonable first step is to hand code a sample that seems to be of the smallest size sufficient to create reliable computer models, given the complexity of the categories. If the size is not sufficiently large, more data can be hand coded at a later date, so we recommend starting with a large data set, but coding subsets until the software is able to build reliable models from the coded data.

The main benefit to LightSIDE is that it is free, open source software easily accessible to all researchers. In addition, LightSIDE learns from the training set so, once the data are hand coded, it is relatively easy to run several models in LightSIDE, varying parameters to find the optimal model, and there is little work that needs to be done by the researcher after that point. Furthermore, the results of the analysis in LightSIDE provide the researchers with a list of features (i.e. *n*-grams) that are associated with responses that are either included or excluded from a category. The use of LightSIDE as a data mining or exploratory data analysis can illuminate features of the data that may not have been apparent in hand coding, which can then be used to refine and develop coding rubrics for previously collected data. For example, when creating the model for the category "Equally Likely," LightSIDE selected the tokens *hat* and *coin*, leading to the creation of the category, "Agents." LightSIDE may be used for exploring the data even without the burden of having hand coded the data. Consider a data set with responses collected pre- and post-instruction. These data can be loaded into LightSIDE and a model can be generated to list the lexical tokens that differentiate between pre- and post-instruction responses. In the case of *random*, we would hope that tokens such as *probability* or *coin* might be associated with post-instruction responses. Information gained from the LightSIDE analysis can also be used to inform the development of future research, both research questions and design of data collection.

A drawback to LightSIDE is that the model creation is a bit of a black box and not easy to adjust when the researcher notices a systematic error, such as occurred in the analysis reported here when LightSIDE failed to detect key words, such as *predict*. In addition, LightSIDE is most stable when working on one category at a time. If a data set contains responses coded into multiple categories, such as in this data set where each response was categorized as in or out of each of the seven categories, there tends to be a need to reload the data when creating each model. The results for each model must be exported separately and then the output files can be combined to create one file for all categories found in the data. Another limitation to LightSIDE is the need for a large hand-coded data set that contains sufficient responses to represent being in and being out of the category being modeled. Related to this is the possibility that tweaking the model tuning parameters when using a small data set can result in over-fitting the model. If one has sufficient hand coded data, the models can be checked for such an over-fit by reserving part of the data set. Otherwise, additional data may need to be collected and hand coded in order to check the model.

The main benefit to TAS is in its flexibility in allowing the user to create libraries and categorization rules. Furthermore, because the user is more involved in the creation of rules and categories, one has an opportunity to learn about the structures of the data, similar to that which occurs in hand coding, but with greater efficiency. In the random data set, for example, creating rules to model "Without Bias, Representative" illuminated the fact that this one category might actually comprise three separate ideas: lack of bias, representative samples, and independence of observations. This observation will be taken into account in the design of future research on student understanding of the word *random*. For example, the sub-categories and their relationships to the use of terms like *probability*, *likelihood*, or *chance* might in the future be used to identify students who have a stronger understanding of random in the statistical sense than the students who invoke equally likely outcomes. Another benefit to TAS is the graphical and output options that are stronger than those found in LightSIDE. One example is the webmaps illustrated by Figures 2 and 3, which allow the researcher to view the connections between the use of terms. TAS also provides tabular information on the number of responses in particular categories and in the overlap between categories. In addition, when a single data set has been coded into multiple categories, the software is able to output a spreadsheet containing a list of the categories into which each subject's response has been placed. The main limitation to the use of TAS is the cost of the license, even with educational pricing. An additional limitation is the time and effort needed on the part of the researcher to create the libraries, categories, and rules in TAS.

5.3 Implications and Future Directions

Even given the limitations of the software, both programs discussed in this paper are a more reasonable approach than hand-coding a great number of student responses should a researcher want to use open-ended responses in order to learn more about student knowledge than is possible through the use of forced response, multiple-choice, or true-false type, questions. In fact, the coding of the random data suggests that the best use of the software may be to employ an iterative process of hand coding in conjunction with the two programs. Once sufficient data have been hand coded to create a reasonable model in LightSIDE, the tokens identified by LightSIDE can be used as a basis for creating libraries and categories in TAS. Researchers can cycle through the three methods until one of the software packages has created a model that produces reasonable kappa values for the data. This model can then be applied to previously un-coded data or newly collected data.

The iterative process is one way in which the software can be used to inform the research process. The underlying research question for which we used the software in this project was to understand how statistics students define and use the word *random*. In the process of coding the data using the software packages, we found that some of the categories we had created by hand might contain several distinct ideas that should have been categorized separately. For example, the category, "Without Bias or Representative," appears to be an amalgamation of three distinct statistical ideas: unbiased (i.e. sampling or assignment to treatments), representative (i.e. samples or treatment groups), and independent (i.e. selection of subjects). The issues with this category were raised by the difficulty LightSIDE had in creating a reasonable model for this category. The knowledge gained about this category will be retained in future studies of student understanding of *random*.

The analysis may also lead to more general directions for research programs. Originally, the hand coding rubric for the random data included two dimensions: the definitional categories reported here and a usage category. The usage category was coded as whether the subject used *random* as a non-specific adjective, as a descriptor of a process, or a descriptor of an outcome (see Kaplan, et al., 2010 for more details). Another possible grouping for the responses is into categories for subjects who wrote specifically about random sampling, random assignment, or neither. Creating these categories would be relatively easy using TAS and the results could inform future research studies about student understanding of random samples in experimental design. This analysis would most likely require revision of the categories because the researcher may be more interested in how or if students describe the sampling process rather than capturing definition types of random with the categories (e.g. "By Chance"). The library created for the project described here could, however, be used as a basis for the subsequent analysis.

One long-term goal of incorporating automated analysis of student responses into statistics education research is to be able to provide real-time feedback to statistics instructors about their students' understanding of statistical concepts. Once the models for student definitions of *random* have been tested on further data and found to be reliable, instructors would be able to upload their own students' responses, which would then be automatically analyzed. The instructors would receive a report documenting the types of definitions provided by students, the number of students who provided each type of definition and the relationships among the tokens mentioned by students, in the form of a webmap or similar graphic. The goal of automatic analysis of student responses is not limited to statistical definitions. In fact, the AACR project is already successfully scoring student responses to biology prompts (for example see: Haudek, et al., 2012; Weston, Parker, and Urban-Lurain, 2013) and is developing models that will analyze student descriptions of histograms and using the software to illuminate common misconceptions about histograms (for more detail, see Kaplan, Gabrosek, Curtiss, et al., 2014).

6. REFERENCES

Abu-Mostafa, Y. S. (2012). Machines that think for themselves. *Scientific American, 307*(1), 78-81.

Aliaga, M., Cobb, G., Cuff, C., Garfield, J., Gould, R., Lock, R., Moore, T., Rossman, A., Stephenson, R., Utts, J., Velleman, P. and Witmer, J. (2005). The Guidelines for Assessment and Instruction in Statistics Education (GAISE) College Report. Endorsed by the Executive Committee of the American Statistical Association. Available at: http://www.amstat.org/education/gaise/

American Association for the Advancement of Science. (2009) Vision and change: A call to action. Author: Washington, DC. pp. 11.

Beggrow, E. P., Ha, M., Nehm, R. H., Pearl, D. & Boone, W. J. (2013). Assessing scientific practice using machine-learning methods: How closely do they match clinical interview performance? *Journal of Science Education and Technology, 23*(1), 160-182.

Bejar, I. I. (1991). A methodology for scoring open-ended architectural design problems. *Journal of Applied Psychology, 76*(4), 522-532.

Bennett R.E. & Ward W.C. (Eds.). (1993) Construction versus choice in cognitive measurement: Issues in constructed response, performance testing, and portfolio assessment. L. Erlbaum Associates: Hillsdale, N.J. pp. xi, 339.

Birenbaum M. & Tatsouka K.K. (1987) Open-ended versus multiple-choice response formats – It does make a difference for diagnostic purposes. *Applied Psychological Measurement, 11*, 329-341.

Bridgeman B. (1992) A comparison of quantitative questions in open-ended and multiple-choice formats. *Journal of Educational Measurement, 29,* 253-271.

Cobb, G. (1992). Teaching Statistics. In L.Steen (ed.), *Heeding the Call for Change: Suggestions for Curricular Action*, pg 3 – 43. Mathematical Association of America: Washington, D.C.

Committee on Undergraduate Science Education. (1999) Transforming undergraduate education in science, mathematics, engineering, and technology National Academy Press: Washington, DC.

Corbin, J., & Strauss, A. (Eds.). (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory (3rd Edition)*. Sage Publications Inc: Thousand Oaks, CA.

D'Avanzo C. (2008) Biology Concept Inventories: Overview, Status, and Next Steps. *Bioscience, 58*, 1079-1085.

Duit R. (2009) Students' and Teachers' Conceptions and Science Education. Accessed March 14, 2011 online at: http://www.ipn.uni-kiel.de/aktuell/stcse/

Fielding-Wells, J. (2014). Where's your evidence? Challenging young students' equiprobability bias through argumentation. In Makar and Gould (eds), *Proceedings of the 9th International Conference on Teaching Statistics*. Flagstaff, AZ. Online at: http://icots.info/9/proceedings/pdfs/ICOTS9_2B2_FIELDINGWELLS.pdf

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*, 378-382.

Gal, I. & Garfield, J. (1997). Curricular goals and assessment challenges in statistics education. In Gal and Garfield (eds), *The Assessment Challenge in Statistics Education,* pp. 1 – 13. IOS press. http://iase-web.org/documents/book1/chapter01.pdf

Garvin-Doxas, K. & Klymkowsky, M.W. (2008) Understanding randomness and its impact on student learning: Lessons learned from building the Biology Concept Inventory (BCI). *CBE Life Sciences Education, 7*(2): 227 – 233.

Gess-Newsome J., Johnston A., & Woodbury S. (2003) Educational reform, personal practical theories, and dissatisfaction: The anatomy of change in college science teaching. *American Educational Research Journal, 4,* pg. 731-767.

Ha, M., and Nehm, R. (2011). Comparative efficacy of two computer-assisted scoring tools for evolution assessment. Paper presented at the *National Association for Research in Science Teaching*. Orlando, FL.

Ha, M., Nehm, R. H., Urban-Lurain, M. & Merrill, J. E. (2011). Applying computerized scoring models of written biological explanations across courses and colleges: Prospects and limitations. *CBE-Life Science Education, 10*, 379-393.

Haudek, K.C., Kaplan, J.J., Knight, J., Long, T., Merrill, J., Munn, A., Nehm, R., Smith, M. Urban-Lurain, M. (2011), Harnessing technology to improve formative assessment of student conceptions in stem: Forging a national network. *CBE Life Sci Educ, 10*, 149-155.

Haudek, K. C., Prevost, L. B., Moscarella, R. A., Merrill, J., and Urban-Lurain, M. (2012), What are they thinking? Automated analysis of student writing about acid-base chemistry in introductory biology. *CBE Life Sci Educ*, *11,* 283-293.

Kaplan, J.J., Fisher, D. & Rogness, N. (2010). Lexical Ambiguity in Statistics: How students use and define the words: association, average, confidence, random and spread. *Journal of Statistics Education, 18(2),* http://www.amstat.org/publications/jse/v18n2/kaplan.pdf

Kaplan, J.J., Gabrosek, J.G., Curtiss, P. & Malone, C. (2014). Investigating student understanding of histograms. *Journal of Statistics Education*, *22*(2).

Kaplan, J.J., Rogness, N. & Fisher, D. (2014). Exploiting Lexical Ambiguity to Help Students Understand the Meaning of *Random*. *Statistics Education Research Journal, 13*(1), 9 – 24. http://iase-web.org/documents/SERJ/SERJ13%281%29_Kaplan.pdf

Kardash, C.A., & Wallace, M.L. (2001) The perceptions of science classes survey: What undergraduate science reform efforts really need to address. *Journal of Educational Psychology, 93*, 199-210.

Knight, J.K. (2010) Biology concept assessment tools: Design and use. *Microbiology Australia 31*, 5-8.

Kuechler W.L. & Simkin M.G. (2010) Why is performance on multiple-choice tests and constructed-response tests not more closely related? Theory and an empirical test. *Decision Sciences Journal of Innovative Education, 8,* 55-73. DOI: 10.1111/j.1540-4609.2009.00243.x.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*, 1159-1174.

Libarkin, J.C. (2008) Concept inventories in higher education science, Council Promising Practices in Undergraduate STEM Education Workshop 2, National Research Council, Washington, DC.

Mayfield, E., Adamson, D., & Rosé, C. (2013). LightSIDE: Researcher's User Manual.

Mayfield, E. & Rosé, C. (2010). An interactive tool for supporting error analysis for text mining. Paper in Proceedings of the Demonstration Session at the International Conference of the North American Association for Computational Linguistics (NAACL), Los Angeles, USA.

Moharreri, K., Ha, M., & Nehm, R. H. (2014). EvoGrader: an online formative assessment tool for automatically evaluating written evolutionary explanations. *Evolution: Education and Outreach, 7*(1), 1 – 14.

National Science Foundation. (1996) Shaping the future: New expectations for undergraduate education in science, mathematics, engineering and technology, National Science Foundation, Directorate for Education and Human Resources, Washington, DC. pp. 76.

Nehm, R. H., Ha, M., & Mayfield, E. (2012). Transforming biology assessment with machine learning: Automated scoring of written evolutionary explanations. *Journal of Science Education and Technology, 21*(1), 183-196.

Nehm, R. H., & Haertig, H. (2012). Human vs. computer diagnosis of students' natural selection knowledge: Testing the efficacy of text analytic software. *Journal of Science Education and Technology, 21*, 56-73.

Nehm, R.H. & Schonfeld, I.S. (2008) Measuring knowledge of natural selection: A comparison of the cins, an open-response instrument, and an oral interview. *Journal of Research in Science Teaching, 45*, 1131-1160.

Ochs, T.L. (1990). Nonrandom uses. *Nature, 343*, 303.

Pellegrino, J.W., Chudowsky, N., & Glaser, R. (2001) Knowing what students know: The science and design of educational assessment National Academy Press: Washington, DC.

Platt, J. C. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MsR-TR-98-14, Microsoft Research.

Ruiz-Primo, M.A., Shavelson, R.J., Hamilton, L. & Klein, S. (2002) On the evaluation of systemic science education reform: Searching for instructional sensitivity. Journal of Research in *Science Teaching, 39*, 369-393.

Seymour, E. (2002) Tracking the processes of change in US undergraduate education in science, mathematics, engineering, and technology. *Science Education 86*, 79-105.

Seymour, E. & Hewitt, N.M. (1997) Talking about leaving: why undergraduates leave the sciences Westview Press, Boulder, Colo.

SPSS (2011), "IBM SPSS Modeler 14.2."

SPSS (2010), SPSS Text Analytics for Surveys 4.0 User's Guide, Chicago, IL: SPSS, Inc.

Tobias, S. (1990) They're not dumb, they're different: Stalking the second tier. 94 ed. Research Corporation: Tucson, AZ.

Wang, H. C., Chang, C. Y., & Li, T. Y. (2008). Assessing creative problem-solving with automated text grading. *Computers & Education, 51*(4), 1450-1466.

Weston, M., Parker, J. M., & Urban-Lurain, M. (2013). Comparing formative feedback reports: Human and automated text analysis of constructed response questions in biology. Paper presented at the *National Association for Research in Science Teaching Annual Conference*. Rio Grande, Puerto Rico.