User's Manual for POES version 2.0.1

Duncan Ermini Leaf
Kimberly A. Barchard

September 9, 2013

Contents

1 Overview

Program for Open-Ended Scoring (POES; Ermini Leaf & Barchard, 2012) is a computer program for automated scoring of open-ended tests. Starting with version 2.0.0, POES is written in the Python programming language. Development and testing of POES used Python version 2.7.1. If Python is not already installed on your system, you will need to install it. This manual contains documentation of how to use POES version 2.0.1.

An Eating Habits Survey was created to illustrate the operation of POES in this manual. This survey will be referred to many times in this manual. More information about the Eating Habits Survey can be found in Appendix A.

To calculate scores, POES takes two files as input. The first file is the subject data file. This contains the response data from your test subjects. Information about the format of this file can be found in the Subject Data File section (3) of this manual. The second file is the wordlist. The wordlist contains a list of words and phrases (keys) that you want POES to look for when scoring. Information about the format of this file can be found in the wordlist file section (4) of this manual.

POES calculates scores using various scoring methods. These methods can compute scores for any of four levels: at the level of the item subpart, at the level of the item, at the level of the subject, and at the level of the batch. A batch is all subjects being scored in a particular run of POES. POES comes with several scoring methods. More information about these methods can be found in the POES Scoring Methods section (5) of this manual. It is also possible to implement new scoring methods as plug-in modules for POES using the Python programming language. Information about scoring module development can be found in the POES Development Manual document.

POES can output the score information in two different formats. First, there is the Score Report Output File. This file contains the responses, the wordlist keys found in the responses, and the scores for each response. More information about the score report output file can be found in the Score Report Output File section (6.1) of this manual. The second type of output file is the Score Data CSV Output File. This file contains comma-delimited scores for each response. It can be easily imported into Excel or SPSS for data analysis. More information about the format of this data output file can be found in the Score Data CSV Output File (6.2) chapter of this manual. It is also possible to create customized output formats using the Python programming language. See the POES Development Manual for details.

Before running POES, you must install the Python interpreter. Information can be found in the Installing Python section (7) of this manual. POES was developed and tested using Python version 2.7.1. You should be able to use POES with any 2.7.x version of Python. Compatibility with Python version 3 or higher has not been tested. Once you have installed Python, have formatted your two input files correctly, and decided which output files you want, you are ready to run POES. Information about how to run POES can be found in the Running POES chapter (8) of this manual.

2 Summary of POES Operation

When you run POES, the basic operation is as follows. First, the input files are read. Next, POES searches the text of each subpart response in the Subject Data File looking for keys contained in the wordlist file. The result for each subpart is a list of wordlist keys that appear in the response and a count of the number of times that each key appears. These key counts are also aggregated at the item level, subject level, and batch level. After the key counts are obtained, POES applies any scoring modules you have specified. Specific scoring modules are discussed later in this document, but in general, a scoring module can operate at any level (subpart, item, subject, batch) using wordlist key counts and values to assign scores. The end result is a table at each level containing the names of scoring methods and the scores that were assigned. Lastly, POES supplies these score tables to one or more output modules. Each output module formats the score data and writes it to an output file.

To determine if wordlist keys are contained in the subject's response, POES looks for the longest match of the response text to the wordlist key. This means that POES starts with the first word in a response and continues word-by-word until it finds a word matching the beginning of a wordlist key. The search continues sequentially until it is determined whether or not the sequence of response words matches an entire wordlist key. If there is *not* a match, POES begins the search again, starting immediately after the word that matched the beginning of a wordlist key. If there *is* a match, POES keeps a record of the match and then continues searching for new matches starting from the response word immediately after the words that matched the wordlist key.

At present, this sequential search of the longest match is the only search algorithm available in POES. Additional search algorithms could be added in the future. See the POES Developer's Manual for more information.

3 Subject Data File (input)

To produce score data, POES requires the response data from your subjects. POES has no limit for the number of subjects in the subject data file. However, attempting to score a very large file could make your computer run slowly or cause POES to exit due to memory limitations. For very large subject data files or older computer systems, you should split your subject data file into smaller files and then score each one separately.

The response data file must be specially formatted so that POES can quickly read the subjects' response data. This section discusses the formatting of the subject data file. In section 3.1 we will discuss the raw format of the subject data file. Then, in section 3.2, we will show how the data can be formatted easily using Microsoft Excel.

3.1 Raw Format of Subject Data File

The Subject Data File must be in comma-separated-value format. In older versions of POES, the first line of the subject data file was required to have the number of subjects in the data file. For backward compatibility, it is still necessary to have an integer on the first line of the file. However, it can be any integer. POES will check to see whether this number matches the actual number of subjects it finds in the file. If it does not match, POES will show a warning message, but continue scoring. It is recommended that you put your intended number of subjects on the first line of the file as a tool for error checking.

Each subsequent line of the subject data file contains data for one subpart. Each data line uses the following format:

$$subject\_ID, item\_ID, subpart\_ID, subpart\_response\_text$$

> $subject\_ID$ – The first element on each line is the subject ID. The file in Figure A has two subjects, identified as 101 and 102. Subject IDs need not be numerical: text identifiers are acceptable. For example "John Smith" could be used as a subject ID. The subject ID is followed by a comma. Any spaces before this comma will be included at the end of the subject ID. Any spaces after this comma will be included at the beginning of the item ID.

item_ID – The second element on each line is the item ID. In Figure A, there are four items. Item IDs can also be any text, excluding commas. The item ID is followed by a comma. Any space before this comma will be included at the end of the item ID. Any space after this comma will be included at the beginning of the subpart ID.

subpart_ID – The third element on each line is the subpart ID. In Figure A, each item has two subparts. Text IDs can be used instead of numbers, but may not contain commas. The subpart ID is followed by a comma. Any space before this comma will be included at the end of the subpart ID. Note that up to this point, there are no spaces adjacent to any comma because we do not want spaces in the subject ID, item ID, or subpart ID.

subpart_response_text – Finally, the last part of each line is the subject's response text for that subpart of that item. The subpart response text may contain commas (with or without adjacent spaces) and any other character combinations. POES assumes that everything after the third comma on a line is the subpart response text. Each data line must be given on a single line in the subject data file. Make sure that there are no line breaks in the middle of the subpart response text. Technically, the number of characters you can put on a data line of the file is limited only by your computer's memory. However, if the subpart response text contains several hundred words, some text editors may not allow you to put the entire response on one line in the file. In that case, you may need to find a different text editor, such as Wordpad in the Microsoft Windows environment.

Figure A contains an example Subject Data File for an eating habits survey. This survey was created by the authors for the sole purpose of demonstrating the use of POES software. This example survey is not intended for actual use in any clinical or experimental setting. More information about this example survey can be found in Appendix A.

Consider the example Subject Data File in Figure A. The first line says that this file is intended to have data for two subjects. POES will verify that this file contains the data for 2 subjects. The second line is the first of the data lines. It says that subject 101's response to Item 1, Subpart 1 is "When I was young I dreamt about cupcakes, ice cream, and squash. I always hated squash." The third line is the second data line. It says that subject 101's response to Item 1, Subpart 2 is "Now I dream about cheescake and tiramisu." This is the third line of the file, even though it is the fourth line in the figure. Although some of the subpart responses appear to be on multiple lines in this figure due to the page width of this document, the actual file must have each datum on a single line. Note that Subject 102 did not provide a response to Item 1, Subpart 2. However, to know that this is a blank response, POES requires the subject, item, and subpart IDs to be specified. If this line of the subject data was completely omitted from the file, POES would proceed as if that subpart had not been administered to the subject. When a subject does not give a response, you can leave the subpart response field blank or use any non-alphabetical character. POES ignores non-alphabetical symbols and so it will see the response as blank.

3.2 Preparing the Subject Data File with Microsoft Excel

Microsoft Excel allows you to save a worksheet in comma-separated-value (CSV) format, which can be used directly with POES. When you save a worksheet in CSV format, Excel will insert a comma between each column, and will put each row on a separate line. To make use of this format, open Excel. For each subpart response, repeat these steps on each row:

1. In the first column, enter the subject ID

2. In the second column, enter the item ID
3. In the third column, enter the subpart ID
4. In the fourth column, enter the subpart response text

Once you have completed entering the data, click the “File” button and choose “Save As.” Enter a name and location for your Subject Data File. You should see a “Save as type:” or “Format:” dropdown menu near the field where you type the filename. From this dropdown menu, choose “CSV (Comma delimited) (*.csv)” or “Comma Seperated Vales (.csv)” and then click the “Save” button. Close your Excel file.

Next, open up your Subject Data File with a text editing application, such as Wordpad in Windows or TextEdit for Mac. Notice that the file conforms to the Raw Format described in section 3, except for the first line. Using your text editor, create a new line at the top of the file, and on that line, type the number of subjects in the file. Save the file. Your file is now ready for scoring by POES.

If you edit the CSV formatted table using Excel, it will add extra commas on the first line, after the number of subjects. This does not match the format of the Subject Data File and will prevent POES from scoring your data. If there are extra commas (or any other characters) after the number of subjects, you will need to open the file again using a text editor like WordPad and remove the extra characters.

3.3 Spell-Checking the Subject Data File

POES does not spell-check data. If a word is spelled incorrectly, it will not match a wordlist key (unless the wordlist includes that specific misspelling of the word). Therefore, it is critical that responses be spell-checked before being scored by POES.

To spell-check the response data using Microsoft Word, use the following steps:

Open the subject data file

Open Microsoft Word

Click on the File menu, and select Open

For Windows systems, make sure that the “Files of type:” menu shows “All files (*.*)” For OS X systems, set the “Enable:” dropdown menu to “All Files.”

Navigate to the CSV file containing the Subject Response Data, and open the file.

Spell-check the responses

Click on the Tools menu, and select “Spelling and Grammar”

When Microsoft Word finds a misspelled word, carefully read the response to determine what word the participant was trying to spell.

Save the file

Click on the File menu, and select “Save”

Microsoft Word will warn you that the file may contain features that are not compatible with plain text format, and ask if you want to save the file in this format. Click Yes.

4 Wordlist File (input)

The wordlist file contains a list of words and phrases (keys) with their associated values. During the scoring process, POES searches subjects' responses looking for these wordlist keys. Scoring methods typically calculate scores using the values for those keys found in the response. It is possible to use non-numerical values as long as they are compatible with the scoring methods. For example, if the AllSum method is used, the addition operation (+) must be defined for all values in the wordlist. POES has not yet been fully tested with non-numeric values, but the functionality is mentioned here for those who are interested.

Figure A.2 shows the contents of an example wordlist file. The general format of the wordlist file is as follows: The first line of the wordlist file is used to provide information about the wordlist, such as the name of the wordlist, the author, and the revision date. This information will be displayed in the two output files.

The second line of the file contains an APA-style reference for the wordlist. This reference will be displayed in the two output files. This reference is given directly in the output so that POES users will know how to cite the wordlist they used when they describe their scoring procedures in publications and presentations. This will minimize confusion about how a set of data was scored and will ensure that the creator of a specific wordlist can be properly credited in publications. Note: This reference must fit on exactly one line. No line breaks can be present.

After the first line, each wordlist key and its value are displayed using three lines:

1. The first line is the word or phrase comprising the key. When POES reads the key, any uppercase letters will be converted to lowercase and any punctuation or non-alphabetical characters will be removed. If punctuation would sometimes be expected in responses, represent the word or phrase as two separate keys: once with a space where the punctuation would occur and once with no space. For example, "open-minded" should be represented as "openminded" and "open minded".

2. The second line is the value associated with the key on the preceding line.

3. The third line is blank. Note that the third line is a blank even for the last Valuable in the list. Therefore, the last line of the file must be blank.

5 POES Scoring Methods

This version of POES comes with several scoring methods that can be used to calculate total scores for each subject. Scoring methods can assign scores to any of the subpart, item, subject, or batch levels. In each method that comes with POES, the same algorithm is applied at each level. Only the inputs to the algorithm change at each level. Additional scoring methods can be implemented as plug-in scoring modules for POES using the Python programming language.

When considering scoring methods it can be helpful to keep in mind how each method treats the values assigned to wordlist keys. Some methods treat the values as ordinal, some treat them as nominal, and some methods do not make use of these values at all.

Some scoring methods use all wordlist keys present in a response while other methods use only unique keys. In general, methods with "Unique" in their name will only use unique keys. These methods are designed to give higher scores to responses that include multiple, distinct wordlist keys than to responses that use the same wordlist keys repeatedly.

Some scoring methods have user-specified parameters in configuration files. The format of each configuration file is discussed in the section for the scoring method. For Windows users, Wordpad is the recommended tool for editing these files. Notepad can cause problems with line breaks that make the files unreadable to POES.

5.1 The CountWords Scoring Method

The CountWords method simply counts the number of words in the response. Unlike most other scoring methods, it does not use the wordlist keys or values in any way. In the CountWords scoring method, subpart scores are the number of words in the subpart. Item scores are the number of words used in all the subparts. Subject scores are computed as the sum of the item scores. The batch score is the sum of the subject scores. Summing at the subject and batch levels is equivalent to applying the CountWords algorithm at each levels. The score at each level is equal to the number of words at that level.

In some settings, the number of words in a response will be interesting in and of itself. The CountWords scores alone will be interesting. Other times, the number of words can be combined with other scores from the response (such as AllSum scores) to calculate a density score (the average score for each word).

5.2 The AllSum Scoring Method

The All-Sum method calculates the subpart, item, subject, and batch scores as the sum of all the values for wordlist keys found in the response. First, subpart scores are calculated as the sum of the values for individual wordlist keys found in the subpart. For example, in the Eating Habits Survey, the first subpart response is: "When I was young I dreamt about cupcakes, ice cream, and squash. I always hated squash." The wordlist keys in this subpart are "cupcakes" (value 3) and "ice cream" (value 3). Therefore, the subpart score is the sum of 3 and 3, or 6. The second subpart response for the first subject is: "Now I dream about cheescake and tiramisu." The only wordlist key in this list is "tiramisu" (value 3), because "cheesecake" is misspelled. Therefore, the subpart score is 3. Next, item scores are calculated as the sum of the values in the aggregate list of wordlist keys found in any subparts of the item. The combined list for the first item for the first subject is "cupcakes" (value 3), "ice-cream" (value 3), and "tiramisu" (value 3), which gives an item score of 9. Third, subject total scores are calculated as the sum of the item scores. In the example given here, we have only shown you how the item score was calculated for the first item. This process would be repeated for all the remaining items, and then the subject score would be the sum of those item scores. Similarly, the batch score is the sum of all of the subjects' scores.

The All-Sum method is the POES scoring method that is most likely to be applicable to scoring a wide variety of open-ended tests. It incorporates all of the information given in a response, and is the most straight-forward of the POES Scoring Methods.

5.3 The AllSumUnique Scoring Method

The AllSumUnique method calculates scores at each level as the sum of the values for all *unique* wordlist keys found at that level. Each subpart score is the sum of unique values for wordlist keys found in that subpart. Item scores are computed as the sum of all values for unique wordlist keys found in any subpart of the item. Subject scores are computed by summing the values for all unique wordlist keys found across every item subpart for a subject. Finally, batch scores are computed by summing all unique wordlist key values across all subjects in the batch.

As an a example in the Eating Habits Survey, the first subpart response contains "ice cream." "Ice cream" also appears in this subject's response to item 2, subpart 1. Therefore, the list of unique wordlist keys for this subject contains "ice cream" (value 3) once.

5.4 The HighestN Scoring Method

The HighestN method calculates the sum of the N highest values for wordlist keys found in the response. Subpart scores are calculated as the sum of the N highest values for the wordlist keys found in the subpart. Item scores are calculated as the sum of the N highest values for all the wordlist keys found in any subpart of the item. Subject scores are calculated as the sum of the N highest values for all the wordlist keys found in any subpart of any item. Batch scores are calculated as the sum of the N highest values for all the wordlist keys across all subjects in the batch.

HighestN is a variation of the All-Sum method, designed to reduce the influence of very long responses on scores. Like the All-Sum method, the HighestN method is very general, and may be useful with a variety of open-ended tests.

To change the value of N, use a text editor to open the HighestN.cfg file in the scoring_modules subfolder of your POES installation folder. The value of N is defined below the line with an "N". Save the HighestN.cfg file when you are done.

By default N is set to 4 for backward compatibility with the Highest-4 method in earlier versions of POES. In previous versions of POES, Highest-4 scores were calculated for each item by summing the values for the 4 keys with the highest values, and then the subject score was the sum of the item-level scores. To implement the method using HighestN, specify that $N = 4$, and then manually calculate the subject score as the sum of the subject's item-level scores. Ignore the subpart scores.

5.5 The HighestNUnique Scoring Method

The HighestNUnique scoring method is similar to the AllSumUnique method, but instead of summing the values for *all* unique wordlist keys at each level, the HighestNUnique score is calculated as the sum of the highest N values. Subpart, item, subject, and batch scores are calculated identically. At all levels, the N keys with the highest values are located and their values summed.

In previous versions of POES, Highest40-AllinOne was calculated for each subject by summing the values for the 40 unique keys with the highest scores across all items. To implement that method using HighestNUnique, specify that N = 40, and use the subject scores. Ignore the subpart and item scores.

The number of values to sum, N, must be specified in the HighestNUnique.cfg file. By default, N is set to 5. To change the value of N, use a text editor to open the HighestNUnique.cfg file in the scoring_modules subfolder of your POES installation folder. The value of N is defined below the line with an "N". Save the HighestNUnique.cfg file when you are done.

5.6 The UniqueMaximums Scoring Method

To calculate scores, the UniqueMaximums method requires five user-specified values: $KeyMax$, $SubpartMax$, $ItemMax$, $SubjectMax$, and $BatchMax$. You only have to specify these values up to the highest level of score that will be meaningful to you. For example, if you do not care about UniqueMaximums scores at the subject and batch levels, then you can use any arbitrary numbers for $SubjectMax$ and $BatchMax$.

The UniqueMaximums scoring method starts at the subpart level and works its way up the levels to end at the batch level. For each subpart, UniqueMaximums starts with a subpart score that is equal to $KeyMax$ or the highest key value in the response, whichever is lower. If there are multiple keys with a value of at least $KeyMax$, the subpart score is incremented by the number of additional keys that have a value of at least $KeyMax$. The final subpart score is the minimum of this computed score and $SubpartMax$.

Next, UniqueMaximums moves to the item level. For each item, it starts with a score that is equal to the maximum score of any subpart. If more than one subpart has a value of $SubpartMax$, then the item score is incremented by the number of additional subparts with a score of $SubpartMax$. The final item score is the minimum of this computed score and $ItemMax$.

The process repeats at the subject level. For each subject, the initial score is the maximum item score. If more than one item has a value of $ItemMax$, the subject score is incremented by the number of additional item scores equal to $ItemMax$. The final subject score is the minimum of this computed score and $SubjectMax$.

Finally, the batch score is computed by taking the maximum subject score and incrementing it by the number of additional subjects with a score equal to $SubjectMax$ (if any). The batch score is capped at $BatchMax$.

The values of $KeyMax$, $SubpartMax$, $ItemMax$, $SubjectMax$, and $BatchMax$ must be specified in the UniqueMaximums.cfg file. The default settings are $KeyMax = 3$, $SubpartMax = 4$, $ItemMax = 5$, $SubjectMax = 6$, and $BatchMax = 7$. To change any of these values, use a text editor to open the UniqueMaximums.cfg file in the scoring_modules subfolder of your POES installation folder. Each value is defined on the line below the line with its name. For example, the value of $KeyMax$ is defined on the line below the line with "KeyMax". Save the UniqueMaximums.cfg file when you are done.

In previous versions of POES, the 334 scoring method calculated subpart scores based upon the number of unique keys with a value of 3 in the subpart. If there was more than one of these keys, then the subpart received a score of 4. Otherwise the subpart score was taken to be the maximum key value in the subpart. Item scores were calculated as the sum of subpart scores and subject scores were summed item scores. The 334 method scores can be computed with the UniqueMaximums method. To produce 334 scores at the subpart level, set $KeyMax$ to 3 and $SubpartMax$ to 4. $ItemMax$, $SubjectMax$, and $BatchMax$ can be ignored. In the output, item scores, subject scores, and batch scores should be ignored. To compute the 334 item and subject scores, you will need to manually sum the UniqueMaximums subpart scores.

In previous versions of POES, the 3345 scoring method assigned subpart scores using the 334 method. Then, the 3345 method looked at the subpart scores and assigned an item score of 5 if two subparts had a score of 4. Otherwise the item score was taken to be the maximum subpart score. Subject scores were summed from item scores. The 3345 scores can be computed with the UniqueMaximums method. To produce 3345 scores at the subpart and item levels, set $KeyMax$ to 3, set $SubpartMax$ to 4, and set $ItemMax$ to 5. $SubjectMax$ and $BatchMax$ can be ignored. In the output, subject scores and batch scores should be ignored. To compute 3345 subject scores, you will need to manually sum the UniqueMaximums item scores.

5.7 The CountCat Scoring Method

The CountCat (count categories) method treats wordlist values as categories, and counts how many of these categories are used in each subpart. For example, if the wordlist contains keys with values of 0, 1, 5, and 10, then there are four distinct values (categories) in the wordlist, and so each subpart could have a score between 0 and 4.

At each level, CountCat uses a two-step procedure to calculate scores. First, it creates a list of all keys at that level and their values (categories). Then, it counts the number of unique values in this list. The scores at each level are equivalent to the number of categories used at each level.

5.8 The CatFreq Scoring Method

The CatFreq (category frequency) method treats wordlist values as categories, and counts how frequently each category is used for each subpart. Thus, it produces multiple scores for each subpart: one score for each distinct value (category) in the wordlist. For example, if the wordlist contains keys with values of 0, 1, 5, and 10, then there are four distinct values (categories) in the wordlist, and so the CatFreq method will create four scores for each subpart. To calculate these subpart scores, CatFreq uses a two-step procedure. First, for each subpart, it creates a list of all wordlist keys and their values (categories). Then it counts the number of times that each value (category) appears. For example, if the subpart response contains six keys with a value of 0, then the FreqValue0 score for this subpart will equal 6. And if the subpart response contains two keys with a value of 10, then FreqValue10 score will equal 2. If there are no keys with a value of 1 or 5, then FreqValue1 and FreqValue5 will both equal 0.

Item scores for each value are computed as the sum of the subpart scores for that value. For example, if Item 1 has three subparts, then the Item 1 FreqValue5 score is equal to the sum of FreqValue5 for subpart 1, FreqValue5 for subpart 2, and FreqValue5 for subpart 3. Similarly, the subject scores for each value are computed as the sum of the item scores for that value, and batch scores for each value are calculated as the sum of the subject scores for that value. The scores at each level are equivalent to the frequencies of keys in that category.

6 Output Files

The format of POES output is defined through report modules. A report module is a Python program that writes scores and other data to a file. This version of POES comes with two report modules. The first module (HR1) produces a human-readable report showing the response, wordlist keys, and scores. This is discussed in section 6.1. The second module (score_data_csv), discussed in section 6.2, puts the score data in a comma separated values (CSV) file to be read by a spreadsheet program such as Excel. It is possible to create modules for new output formats using the Python programming language. This is discussed in the POES Developer's Manual.

To tell POES that you would like output from specific reporting modules, use the "-r" command line argument followed by a comma-separated list of module names. For example, if you would like output from both the HR1 module and the score_data_csv module, you can use the command line argument "-r HR1,score_data_csv" (without quotes). In previous versions of POES, the HR1 report was requested with the "-p" command line argument and the score_data_csv file could be requested with the "-d" command line option. These options are still available in the new version of POES. For more information about command line arguments, see section 8.

All output files will be saved in the *output* subfolder of the directory from which POES is run. **If this folder does not exist, it must be created before running POES**. The output file names for each module have form [report module name]_YYYYMMDD_hhmmss where YYYYMMDD and hhmmss are the date and time when POES was run. In addition to report modules output files, POES will also copy the configuration files of the scoring methods used. This can be helpful if you want to check which parameter settings were used to produce the score module output. The configuration files can be found in the *output* subfolder with names of the form [scoring module name]_YYYYMMDD_hhmmss.cfg. YYYYMMDD and hhmmss will match the date and time for the corresponding report module output.

6.1 HR1 Report Output File

This section describes the HR1 (Human-Readable report format, version 1) output module. It outputs response and score data in a format that can be helpful for understanding how POES reads responses and assigns scores. In previous versions of POES, this report was requested by using the "-p" command line argument. In the current version, report modules are specified in a list after the "-r" command line argument. You can still use the "-p" option to request this report. However, the filename you specify after "-p" **will be ignored**. The location of output files is discussed in section 8.

Figure A.3 shows an excerpt of the score report that POES generated using the Subject Data File in Figure A and the wordlist in Figure A.2.

The first line displays information about the version of POES that created the score report. Next is the APA-style reference to POES. The line after that displays the first line of the wordlist file. That line can be used to provide information about the wordlist, such as the name of the wordlist, the author, and the revision date. After the wordlist information is the APA-style reference for the wordlist. The next line gives the location of the Subject Data File used to generate the score report. The next line lists the actual number of subjects that POES found in the Subject Data File. The next two lines list the maximum number of items that POES found over all of the subjects, and the maximum number of subparts POES found within any single item.

The data for each subject is displayed as follows. The first line for each subject is the Subject ID. In Figure A.3, the first subject's ID is 101. Below this, the score data for each item is displayed. For each test item, the response data for each subpart is displayed along with the scores for that subpart. After the data for each subpart in this item, the item scores are printed under the "ITEM SCORES" heading. If there is a second item for this subject, then below the score data for the first item, you will see the heading for the next item. The HR1 report does not display batch-level scores.

Let's look at the different parts of the score report more closely. For each item, the score report has two sections: one section for subparts' scores and one section for the item scores. The subparts data section is given first. Under the "SUBPART 1" heading, the text of the subject's response is printed out. Although the subject may have used capitalization and punctuation, POES strips out this extraneous information to facilitate scoring. The response text displayed here is exactly what POES is scoring. Next comes a table of the wordlist keys that POES found in the response. Each key is followed by its value and the number of occurrences in the response. Compare the wordlist in Figure A.2 to the Subpart 1 response in Figure A.3. You can see that the wordlist keys in the response were matched to those in the wordlist. Notice that the response for Subpart 2 contains the misspelled word, "cheescake." The wordlist in Figure A.2 contains the word "cheesecake," which has a value of 3, but does not contain "cheescake." Therefore "cheescake" does not receive a score. Remember that POES does not spell-check the response text. After the list of wordlist keys is the "Scores" heading. In this section, each row contains the name of a Scoring Method followed by the score calculated by that method. See section 5 of this manual for information on the Scoring Methods used by POES.

After the subparts data is the heading "ITEM SCORES". This section begins with a combined table of all the wordlist keys found in every subpart of this item. If you look at the wordlist key count lists from Subpart 1 and Subpart 2 of Figure A.3, you will see that the counts in the item-level table are a combination of the two subpart tables. Under the "Scores" heading are the item scores produced by each Scoring Method. Some scoring methods calculate scores based upon the combined list of wordlist keys immediately preceding, while other methods calculate scores based upon the individual subpart scores given previously. See section 5 for information on the calculation of subpart and item scores for each of the Scoring Methods.

After all of the item data has been displayed for a subject, a list of all the wordlist keys used by this subject appears under the heading: “All unique wordlist keys for subject X (across all items),” where X is the subject ID. Some scoring methods use this list to calculate subject scores. Following this list are the scores for that subject.

6.2 Score Data CSV Output File

This section describes the Score Data CSV File that POES generates. The Score Data CSV Output File is comma-delimited, and can be easily imported into Microsoft Excel and SPSS. The module that produces this report is called score_data_csv. In previous versions of POES, this report was requested by using the “-d” command line argument. In the current version, report modules are specified in a list after the “-r” command line argument. You can still use the “-d” option to request this file. However, the filename you specify after “-d” **will be ignored**. The location of output files is discussed in section 8.

Figure A.4 shows the Score Data CSV Output File that POES generated using the Subject Data File in Figure A and the wordlist in Figure A.2.

The first four lines of the Score Data CSV Output File provide information about POES and the wordlist. The first line displays information about the version of POES that created the score data file. The next line displays an APA-style reference for POES. The following line displays the first line of the wordlist file. That line can be used to provide information about the wordlist, such as the name of the wordlist, the author, and the revision date. Following the wordlist information line is an APA-style reference for the wordlist. The line below that gives the location of the Subject Data File used to generate the score data. The following three lines list the actual number of participants in the Subject Data File, the maximum number of items for any subject, and the maximum number of subparts for any item across all subjects.

The next row of the file is a list of variables labels, separated by commas. The first column label is subjID. This column contains each subject’s ID. The next set of columns gives the names of the scores for each item and subpart, for each scoring method. The last set of columns gives the subject level scores for each scoring method. When this data is imported into Microsoft Excel or SPSS, the variable labels will become column headers, conveniently labeling all of the scores.

The Subpart Score labels use the following convention:

$$iXsYmZ$$

where X = the item ID, Y = the subpart ID, and Z = the name of the Scoring Method used to calculate the score. This label can be read as: “Item X, subpart Y, method Z.” For example “i2s1mHighestN” would be read as “Item 2, subpart 1, method HighestN.”

The Item Score labels use the following convention:

$$iXmZ$$

where X = the item number, and Z = the name of the Scoring Method used to calculate the score. This label can be read as: “Item X, method Z.” For example, “i2TmHighestN” would be read as “Item 2, method HighestN.”

The last labels on this row are the subject-level scores for each scoring method. They follow the convention:

$$mZ$$

where Z is the method used to compute the subject score.

After the row that gives the variable labels, each subsequent row contains the score data for a single subject.

To import the Score Data CSV Output File into Excel, follow these steps:

Open Excel

Click on the File menu, and select Open

Make sure that the “Files of type:” menu shows “All files (*.*)”

Navigate to the Score Data CSV Output File, and click on “Open”

A “Text Import Wizard” window will appear. Click the radio button next to “Delimited” and click “Next.”

In the “Delimiters” box, check the box next to “Comma” and uncheck all other boxes

Click “Finish.”

You should now see the score data in a spreadsheet. The first rows of data include the POES version, a reference for POES, wordlist information and a reference, and the input subject data file used for scoring. The following rows of data include the number of subjects, maximum number of items in any subpart, and maximum number of subparts in any item. Finally, variable labels are contained on row 21. Subject data is contained on subsequent rows. Each subpart, item, and subject score will be given in a separate column. Note, this output file does not report batch scores.

Save the file as an Excel file. Click File, Save. Set “Save as type:” to “Microsoft Office Excel Workbook (*.xls)”

To Import the Score Data CSV Output File into SPSS, follow these steps:

First import the data into Excel, and save the file.

Delete the rows giving the POES version and the wordlist information. The first row should give the variable labels. Save the file.

Close the Excel file

Open SPSS

Click on the File Menu, and select Open, Data

Under “Files of type:” select “Excel (*.xls)”

Navigate to the Excel version of the Score Data CSV Output File, and click on “Open”

Click on the box next to “Read variable names from the first row of data”, so that a checkmark appears in this box, and click OK.

The data will now appear in the SPSS data view window. The variable labels will be automatically displayed as column headers. The data for each subject will be given in one row of the file.

Save the file as an SPSS data file.

7 Installing Python

POES requires the Python interpreter in order to run. POES 2.0.1 was tested with Python version 2.7.1. Later Python versions might also work, but POES has not been tested with these versions. Many computer systems already have Python installed. Operating systems such as GNU/Linux and OS X will most likely have Python. To check if Python is installed, try searching your programs for "python" or try typing "python" at the command line. If someone else is responsible for administering your computer system, you could also ask them about the availability of Python. If you have determined that you do not have Python on your computer and would like to install it yourself, go to `http://python.org/download/`. Download the appropriate installation file for your operating system and follow the installation instructions. You might need to restart your computer after installation.

8 Running POES

POES is run from the command line. Previous versions of POES included a graphical user interface for Windows called WinPOES. This has been discontinued. A graphical POES interface is not currently available.

In order to run POES, you must specify the names and locations of the subject data file and the wordlist file. Optionally, you can also specify the number of items and the number of subparts per item. These optional items can be used for error checking to make sure that every subject in your data file has the correct number of items and every item has the correct number of subparts.

Depending on your operating system and your Python installation, you might be able to start the Python interpreter by simply typing "python" at the command line. If this does not work, then you will need to type the complete path to Python. For example, if you are using the Windows operating system and the Python interpreter was installed into the C:\Python27 folder, then you can start the interpreter by typing "C:\Python27\python" at the command line. In what follows, it is assumed that Python was installed into C:\Python27 folder. This might be different on your system. You will need to substitute your installation folder in place of C:\Python27 in the examples below.

When you are running POES from the command line, you must specify the names and locations of the necessary files on the command line itself. The first word on the command line will be something like "C:\Python27\python," which indicates that the rest of the information on the line is a Python program. The second word on the line is the name of the Python program you want to run: "poes.py" After that you will type a series of instructions (called arguments), to tell POES the file names and, optionally, the number of items on the test and number of subparts per item. Table 8.1 lists the command line arguments accepted by POES. Only the "-i" argument is required. Note that if you are using files with spaces in their names, you must enclose the entire file name in quotation marks. See Example 8.2 below.

Table 8.1: POES Command Line Arguments

Argument	Description	Default value
-i filename	Location and name of subject data file	None
-w filename	Location and name of wordlist file	"wordlist.txt" (in the current folder)
-d	Legacy argument that requests the score data CSV output file	None
-p	Legacy argument that requests the HR1 output file	None
-n #	Number of items on the test	None
-s #	Number of subparts per item	None
-m LIST	Comma-separated list of scoring modules to use for producing output files	All of them
-r LIST	Comma-separated list of report modules to use for producing output files	All of them

Example 8.1. *Running POES for human-readable output*

```
C:\Python27\python poes.py -i c:\data\test1.txt -w c:\MyFolder\words1.txt -r hr1 -n 33 -s 5
```

C:\Python27\python	*I want to use the Python interpreter installed in the C:\Python27 folder*
poes.py	*I want Python to run the poes program*
-i c:\data\test1.txt	*The subject response data is in c:\data\test1.txt*
-w c:\MyFolder\words1.txt	*The wordlist is located in c:\MyFolder\words1.txt*
-r HR1	*I want the human-readable score report from the HR1 report module*
-n 33	*I expect there to be 33 subjects*
-s 5	*I expect there to be 5 subparts per item*

Note that there must be a space between arguments, and between each argument and its value. Because the "-m" argument was not specified, POES will use every scoring method in the scoring_modules folder.

Example 8.2. *Running POES for CSV output*

```
C:\Python27\python poes.py -i "subjects data.txt" -m AllSumUnique,HighestN -r score_data_csv
```

C:\Python27\python	*I want to use the Python interpreter installed in the C:\Python27 folder*
poes.py	*I want Python to run the poes program*
-i ”subjects data.txt”	*The subject response data is in the same folder as the poes program, and the filename is called* ***subjects data.txt***
-m AllSumUnique,HighestN	*I want to use the AllSumUnique and HighestN scoring methods.*
-r score_data_csv	*I want the CSV output from the score_data_csv report module*

Note that the list of scoring methods for the "-m" argument must be separated only by commas. There cannot be any spaces in the list of scoring methods. Because the command line does not say where the wordlist is, POES will assume it is in the same location as as the POES program you are trying to run, and that the file name is wordlist.txt.

To run POES from the command line on a Windows system, follow these steps: NOTE: Quotation marks are used for illustration here. Only type what appears inside the quotation marks and not the actual quotation marks.

1. Find the location of the Python interpreter on your computer

   - Click on the Start Button
   - Type ”python” in the search box (in Windows 7) or navigate the Start menu until you find the Python application folder
   - Hold the mouse cursor over the icon for Python
   - After a few seconds, a little box should appear with the full path, for example, “C:\Python27\python.exe.” If that box doesn’t appear, try right clicking on the Python icon and selecting Properties. Somewhere in the resulting dialog window you should see the path to Python.
   - Substitute your path to Python instead of “C:\Python27” in what follows.

2. Open up a command line window

   (a) Click the Start Button

   (b) Navigate to All Programs, Accessories, then Run

   (c) Type “cmd” into the box

   (d) Click “OK”

   (e) You should see a black window with some text in it, and a blinking cursor.

3. Change to the POES folder

   (a) If you installed POES into the C:\POES\ folder, then type: “cd\POES” and press Enter.

   (b) The last line on the screen should read “C:\POES\>” or something similar.

4. Execute POES

   (a) Type “C:\Python27\python poes.py” followed by the appropriate command line arguments (see Table 8.1).

   (b) When POES is finished, the last line on the screen will read “C:\POES\>’ or something similar.

5. Close the command line window.

   (a) Click on the X symbol at the top right corner of the window.

   (b) You can now open your output file(s). See section 6 for information about output files.

A Eating Habits Survey

The following eating habits survey was created by the authors for the sole purpose of demonstrating the use of POES software. This example survey is not intended for actual use in any clinical or experimental setting.

This Eating Habits Survey measures subjects' junk food eating habits. There are four items. Each item has two subparts. Note that the survey has been formatted so that there is a clear distinction between the subparts of the subject's response for each question. Without this, it would be difficult to determine which part of the response belongs to which subpart.

Assuming that you have Python installed in C:\Python27\, the output in Figures A.3 and A.4 could be created by running POES with the command:

```
C:\Python27\python poes.py -i "../sample_files/sample_test_data.txt" -w "../sample_files/sample_wordlist.txt"
```

Eating Habits Survey

Question 1:
What foods did you dream about when you were young?

What foods do you dream about now?

Question 2:
What were your favorite foods when you were young?

What are your favorite foods now?

Question 3:
What foods did you eat regularly when you were young?

What foods do you eat regularly now?

Question 4:
What foods did your parent(s) buy at the grocery store when you were young?

What foods do you buy at the grocery store now?

Figure A.1: Example Subject Data File

```
2
101,1,1,When I was young I dreamt about cupcakes, ice cream, and
    squash.  I always hated squash.
101,1,2,Now I dream about cheescake and tiramisu.
101,2,1,My favorite foods as a child were ice cream, doughnuts, and
    spaghetti.
101,2,2,My favorite foods now are cheesecake and spaghetti.
    Sometimes I like to get a hamburger with a milkshake.
101,3,1,My mom usually made a meat and potatoes kind of dinner, but
    as I got older she made me eat more vegetables.
101,3,2,When I am on a diet, I follow it pretty strictly.  But when
    I quit the diet, I tend to eat a lot of junk foods.
101,4,1,My mom used to get a lot of TV dinners. After she took some
    cooking classes, she used to buy more vegetables, like asparagus
    and cabbage and stuff.
101,4,2,I don't really buy TV dinners anymore.  I buy the diet
    brands.  I usually buy cheese, bread, chicken, some fruits, and
    many times, i must admit, doughnuts ice cream, and soda.
102,1,1,I've never really had a dream about foods.
102,1,2,
102,2,1,When I was young I liked the lasagna that my mom cooked.  I
    also liked it when we went to mexican restaurants.  I like fish
    tacos.  My mom also made excellent pies and cakes.
102,2,2,I guess my favorite foods now are sushi. I like pretty much
    all japanese foods.  I also still like mexican food.
102,3,1,My dad used to be very creative in the kitchen.  He would
    make all sorts of dishes that I can't remember.  I didn't really
    like his cooking.  I preferred my mother's cooking.  The best
    part of dinner was always dessert.
102,3,2,I eat out a lot.  Sushi, mexican food.  Sometimes when I am
    sad I like to binge on cake or ice cream.
102,4,1,A whole lot of different foods.
102,4,2,I don't do much grocery shopping because I eat out a lot.
    Usually I'm at the grocery store to buy dessert items such as
    cakes and ice cream and pastries.
```

Figure A.2: Example Wordlist File

```
Wordlist for the Eating Habits Survey, Duncan Ermini Leaf, January 2013
Ermini Leaf, D. (2013). Wordlist for the eating habits survey. File to be used with POES for scoring of the Eating
    Habits Survey.
cheese cake
3

cheese cakes
3

cheesecake
3

cheesecakes
3

cupcake
3

cupcakes
3

doughnut
3

doughnuts
3

hamburger
1

hamburgers
1

ice cream
3

icecream
3

junk food
1

junk foods
1

milk shake
1

milk shakes
1

milkshake
1

milkshakes
1

pastries
3

pastry
3

pastrys
3

soda
3

sodas
3

tiramisu
2

tv dinner
2

tv dinners
2
```

Figure A.3: Excerpt of Example POES Score Report File

```
Generated by POES Version 2.0.1

APA style reference for POES:
Ermini Leaf, D. & Barchard, K.A. (2013).  Program for Open-Ended Scoring version 2.0.1.  [Unpublished computer program].  Available from Kim Barchard,
Department of Psychology, University of Nevada, Las Vegas, 4505 S. Maryland Parkway, P.O. Box 455030, Las Vegas, NV, 89154-5030, USA, barchard@unlv.nevada.edu

Wordlist Information:
Wordlist for the Eating Habits Survey, Duncan Ermini Leaf, June 2009

APA style reference for Wordlist:
Ermini Leaf, D. (2009). Wordlist for the eating habits survey. File to be used with POES to allow scoring of the Eating Habits Survey.

Subject response data file name:
../sample_files/sample_test_data.txt

Number of subjects: 2

Max. number of items: 4

Max. number of subparts: 2

Subject 101 --------------------------------------------------------
Item 1
  SUBPART 1
    when i was young i dreamt about cupcakes ice cream and
    squash i always hated squash
  Wordlist key     value   num. occurences
    ice cream          3                1
    cupcakes           3                1
  Scores:
    All-Sum: 6
    AllSumUnique: 6
    CatFreq1: 0
    CatFreq2: 0
    CatFreq3: 2
    CountCat: 1
    CountWords: 16
    HighestN: 6
    HighestNUnique: 6
    UniqueMaximums: 4

  SUBPART 2
    now i dream about cheescake and tiramisu
  Wordlist key     value   num. occurences
    tiramisu           2                1
  Scores:
    All-Sum: 2
    AllSumUnique: 2
    CatFreq1: 0
    CatFreq2: 1
    CatFreq3: 0
    CountCat: 1
    CountWords: 7
    HighestN: 2
    HighestNUnique: 2
    UniqueMaximums: 2

  ITEM SCORES:
  Wordlist key     value   num. occurences
    cupcakes           3                1
    tiramisu           2                1
    ice cream          3                1
  Scores:
    All-Sum: 8
    AllSumUnique: 8
    CatFreq1: 0
    CatFreq2: 1
    CatFreq3: 2
    CountCat: 2
    CountWords: 23
    HighestN: 8
    HighestNUnique: 8
    UniqueMaximums: 4
```

<output omitted>

```
All unique wordlist keys for subject 101 (across all items):
Wordlist key     value   num. occurences
  tv dinners         2                2
  ice cream          3                3
  junk foods         1                1
  soda               3                1
  cupcakes           3                1
  cheesecake         3                1
  hamburger          1                1
  tiramisu           2                1
  doughnuts          3                2
  milkshake          1                1

Scores for subject 101:
Scores:
  All-Sum: 33
  AllSumUnique: 22
  CatFreq1: 3
  CatFreq2: 3
  CatFreq3: 8
  CountCat: 3
  CountWords: 160
  HighestN: 12
  HighestNUnique: 15
  UniqueMaximums: 4
```

Figure A.4: Example POES Score Data CSV Output File

```
"Generated by POES Version 2.0.1"

"APA style reference for POES:"
"Ermini Leaf, D. & Barchard, K.A. (2013).  Program for Open-Ended Scoring version 2.0.1.  [Unpublished computer program].  Available from Kim
    Barchard, Department of Psychology, University of Nevada, Las Vegas, 4505 S. Maryland Parkway, P.O. Box 455030, Las Vegas, NV, 89154-5030, USA,
    barchard@unlv.nevada.edu"

"Wordlist Information:"
"Wordlist for the Eating Habits Survey, Duncan Ermini Leaf, June 2009"

"APA style reference for Wordlist:"
"Ermini Leaf, D. (2009). Wordlist for the eating habits survey. File to be used with POES to allow scoring of the Eating Habits Survey."

"Subject response data file name:"
"../sample_files/sample_test_data.txt"

"Number of subjects:",2

"Max. number of items:",4

"Max. number of subparts:",2

subjID,i1s1mAll-Sum,i1s1mAllSumUnique,i1s1mCatFreq1,i1s1mCatFreq2,i1s1mCatFreq3,i1s1mCountCat,i1s1mCountWords,i1s1mHighestN,i1s1mHighestNUnique,
    i1s1mUniqueMaximums,i1s2mAll-Sum,i1s2mAllSumUnique,i1s2mCatFreq1,i1s2mCatFreq2,i1s2mCatFreq3,i1s2mCountCat,i1s2mCountWords,i1s2mHighestN,
    i1s2mHighestNUnique,i1s2mUniqueMaximums,i1mAll-Sum,i1mAllSumUnique,i1mCatFreq1,i1mCatFreq2,i1mCatFreq3,i1mCountCat,i1mCountWords,i1mHighestN,
    i1mHighestNUnique,i1mUniqueMaximums,i2s1mAll-Sum,i2s1mAllSumUnique,i2s1mCatFreq1,i2s1mCatFreq2,i2s1mCatFreq3,i2s1mCountCat,i2s1mCountWords,
    i2s1mHighestN,i2s1mHighestNUnique,i2s1mUniqueMaximums,i2s2mAll-Sum,i2s2mAllSumUnique,i2s2mCatFreq1,i2s2mCatFreq2,i2s2mCatFreq3,i2s2mCountCat,
    i2s2mCountWords,i2s2mHighestN,i2s2mHighestNUnique,i2s2mUniqueMaximums,i2mAll-Sum,i2mAllSumUnique,i2mCatFreq1,i2mCatFreq2,i2mCatFreq3,
    i2mCountCat,i2mCountWords,i2mHighestN,i2mHighestNUnique,i2mUniqueMaximums,i3s1mAll-Sum,i3s1mAllSumUnique,i3s1mCatFreq1,i3s1mCatFreq2,
    i3s1mCatFreq3,i3s1mCountCat,i3s1mCountWords,i3s1mHighestN,i3s1mHighestNUnique,i3s1mUniqueMaximums,i3s2mAll-Sum,i3s2mAllSumUnique,i3s2mCatFreq1,
    i3s2mCatFreq2,i3s2mCatFreq3,i3s2mCountCat,i3s2mCountWords,i3s2mHighestN,i3s2mHighestNUnique,i3s2mUniqueMaximums,i3mAll-Sum,i3mAllSumUnique,
    i3mCatFreq1,i3mCatFreq2,i3mCatFreq3,i3mCountCat,i3mCountWords,i3mHighestN,i3mHighestNUnique,i3mUniqueMaximums,i4s1mAll-Sum,i4s1mAllSumUnique,
   i4s1mCatFreq1,i4s1mCatFreq2,i4s1mCatFreq3,i4s1mCountCat,i4s1mCountWords,i4s1mHighestN,i4s1mHighestNUnique,i4s1mUniqueMaximums,i4s2mAll-Sum,
   i4s2mAllSumUnique,i4s2mCatFreq1,i4s2mCatFreq2,i4s2mCatFreq3,i4s2mCountCat,i4s2mCountWords,i4s2mHighestN,i4s2mHighestNUnique,i4s2mUniqueMaximums,
   i4mAll-Sum,i4mAllSumUnique,i4mCatFreq1,i4mCatFreq2,i4mCatFreq3,i4mCountCat,i4mCountWords,i4mHighestN,i4mHighestNUnique,i4mUniqueMaximums,
   mAll-Sum,mAllSumUnique,mCatFreq1,mCatFreq2,mCatFreq3,mCountCat,mCountWords,mHighestN,mHighestNUnique,mUniqueMaximums
101,6,6,0,0,2,1,16,6,6,4,2,2,0,1,0,1,7,2,2,2,8,8,0,1,2,2,23,8,8,4,6,6,0,0,2,1,12,6,6,4,5,5,2,0,1,2,18,5,5,3,11,11,2,0,3,2,30,10,11,4,0,0,0,0,0,0,22,
    0,0,0,1,1,1,0,0,1,26,1,1,1,1,1,1,0,0,1,48,1,1,1,2,2,0,1,0,1,28,2,2,2,11,11,0,1,3,2,31,11,11,4,13,11,0,2,3,2,59,11,11,4,33,22,3,3,8,3,160,12,15,4
102,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0,0,6,6,0,0,2,1,34,6,6,4,0,0,0,0,0,0,21,0,0,0,6,6,0,0,2,1,55,6,6,4,1,1,1,0,0,1,40,1,1,1,
    6,6,0,0,2,1,22,6,6,4,7,7,1,0,2,2,62,7,7,4,0,0,0,0,0,0,6,0,0,0,10,10,1,0,3,2,30,10,10,4,10,10,1,0,3,2,36,10,10,4,23,16,2,0,7,2,161,12,15,4
```