Using a Trainable Pattern-Directed Computer Program to Score Natural Language Item Item Responses

Randy M. Kaplan

April 1992

GRE Board Professional Report No. 89-19P
ETS Research Report 91-31
Using a Trainable Pattern-Directed Computer Program
to Score Natural Language Item Responses

Randy M. Kaplan

GRE Board Report No. 89-19P

April 1992

This report presents the findings of a research project funded by and carried out under the auspices of the Graduate Record Examinations Board.

Educational Testing Service, Princeton, N.J. 08541
The Graduate Record Examinations Board and Educational Testing Service are dedicated to the principle of equal opportunity, and their programs, services, and employment policies are guided by that principle.

Graduate Record Examinations and Educational Testing Service are U.S. registered trademarks of Educational Testing Service; GRE, ETS, and the ETS logo design are registered in the U.S.A. and in many other countries.

Copyright © 1992 by Educational Testing Service. All rights reserved.
Abstract

In this study we investigated a computer scoring procedure for constructed-response items whose responses consist of natural language phrases or sentences. The costs for scoring such items either manually or by a traditional natural language computer processing approach is typically expensive.

The purpose of this study was to investigate the feasibility of an alternate method to the traditional natural language processing approach. The two main goals of the study were to create a prototype scoring program based on an appropriate methodology and to carry out a preliminary comparison of the results of the scoring methodology to the scores of human raters.

Our approach to scoring is based on the assumption that a set of responses can be described as a small language. From this language we can construct a grammar that describes the language. The grammar can then be used as a tool for recognizing sentences in the language.

With respect to the task of scoring, this means creating a grammar from a sample of responses that have been scored correct or incorrect. This grammar is then used by a program to classify responses that were not used to create the grammar. The program automatically classifies a response as correct (or incorrect).

The computer program was used to score sets of responses from three items. Two sets of data consisted of short responses (3 to 5 words). The third set of data consisted of longer responses (12 to 15 words). The scoring program rating agreed perfectly with those of human raters for the first two data sets and achieved a maximum agreement of 90% for the third data set.

We conclude from these preliminary results that a trainable pattern-directed computer scoring program may be a viable, cost-effective approach to scoring short natural language responses. Longer responses present a number of problems that require further investigation. We suggest a number of ways that the prototype program can be extended to compensate for these more complex responses.
Table of Contents

Introduction ..................................................................................................................1

A Prototype Scoring Program .........................................................................................3
  Approach ......................................................................................................................3
  Program Description ..................................................................................................6

Preliminary Analyses of Program Functioning ...............................................................9
  Method .........................................................................................................................9
  Results .........................................................................................................................12
  Discussion ...................................................................................................................14

Summary and Conclusions ...........................................................................................17

Appendix A
  Prototype Program Training Transcript ......................................................................25

Appendix B
  Reading Passage for Pickleweed Item ..........................................................................37

Appendix C
  Sample of Scored Pickleweed Responses ................................................................41
Introduction

This project began because of our interest in applied natural language processing and also as an extension of previous work of Carlson and Ward (1988), which examined the possibility of analyzing the responses to formulating hypothesis (FH) items with an automated system.

Natural language processing has rarely been applied in practice. Typically, natural language research is knowledge intensive, complex, time-consuming, and costly, so few examples of practical applications exist. Scoring natural language responses to examination items offers a constrained domain in which to explore practical issues of natural language processing. The investigation of these issues also supports the idea that constructed-response items in general and natural language free-response items specifically will play a more prominent role in examinations over the next several years given that these items offer certain advantages over the classical multiple-choice format item (Bennett, 1990).

In the Carlson and Ward (1988) study, a formulating hypothesis item consisted of a brief description of a psychological investigation, a figure or table presenting the data from the study, and a short statement of an important finding. With such an item, the examinee is asked to think of a hypothesis that he or she believes is most likely to explain the finding. In addition to that hypothesis, the examinee is also asked to state other competing hypotheses that should be considered. The responses are expressed as one or more sentences and are scored manually by reading the examinee's hypotheses. The study sought alternatives to this manual scoring.

In Carlson and Ward's (1988) study, a full range of natural language analysis procedures were examined. These procedures included pattern matching, syntactic parsing, and semantic analysis. The study achieved some success with pattern matching, but ultimately the results were not satisfactory (p. 28), chiefly because the number of patterns needed for scoring would be enormous, making it difficult to specify all the possible patterns for an item.

Syntactic analysis and semantic analysis have a disadvantage because they require large and complex amounts of information. In the case of the semantic analysis case frame approach, the contents of the case frames are particularly difficult to specify.

The Carlson and Ward (1988) study points out that each of the methodologies for processing natural language responses has advantages and disadvantages. Given that this is the case, we would like to step back for a moment and consider what the requirements should be for an automatic procedure for scoring natural language responses. We believe the following points are important considerations for an automated scoring system for natural language responses.

1. The scoring procedure should be easy to set up (minimal amount of time necessary to create a scoring key).

Regardless of the scoring procedure used, it will be necessary to specify what constitutes a correct answer and what constitutes an incorrect answer. If an answer was scored on a more complex scale, say, 0 to 5, the specification of what constitutes a particular score would also have to be made. If the approach taken to scoring is knowledge based (specification of a representation of the concepts that constitute a correct answer), the cost (actual time and effort) to specify a correct answer could be quite high. Constructing knowledge bases for individual items could potentially be more costly than scoring the responses manually. Although this
aspect alone may not be enough to rule out automatic knowledge based scoring, it certainly would be a disadvantage.

On the other hand, automating the scoring process may have certain substantial advantages. For example, it would be highly desirable for the scoring procedure to assist in the process of defining a correct answer. An example of this would be a scoring procedure that could construct its own scoring key from a set of responses. Another advantage of an automated scoring procedure is that it can be more consistent than a human rater in assigning scores to a response.

(2) The scoring procedure should be able to score responses on various scales (correct/incorrect, 0 - 5, etc).

Given that in some cases it might be necessary to score a response to an item as correct or incorrect or on a scale of 0 to 5, where 0 represents completely wrong and 5 represents completely correct, the scoring procedure should be able to score on the desired scale.

(3) The scoring procedure should be able to identify those responses that it cannot score.

In conjunction with the second requirement, a scoring procedure should be able to identify any responses that it cannot score. These responses should be set aside to be scored by a human rater. The results of the human assessment of these unscorable responses can be fed back to the scoring program to increase its ability to recognize and score a greater number of responses.

(4) The scoring procedure should be able to "learn" from its mistakes.

The scoring procedure should be able to modify itself to be able to recognize and score an increasingly larger set of responses the more it is used. We envision libraries of descriptions that are associated with items that enable responses to be scored. As more responses to an item are scored, the response library for that item increases.

(5) The scoring procedure should be consistent.

Given the same response more than once, the scoring procedure should score that response the same each time, regardless of whether the response is part of one set of responses or distributed over several sets of responses.

(6) The results of the scoring procedure should be acceptable to human raters.

If the scoring procedure is to be used regularly, the scores it produces must be acceptable to human raters. This includes the case where a human rater may not agree with the score produced by the automatic scoring procedure, yet finds the score acceptable. For example, in the third data set analyzed in this study, one of the responses scored was "Food that can be grown in saltwater." For this response, the human raters did consistently agree as to whether it
was correct or incorrect. Either score could have been assigned by a human rater. In the event of a machine scoring of this response, either score should be acceptable to a human rater.

In the analyses described in this paper, we use the scores of human raters as the criteria for judging the performance of the prototype scoring program. For this investigation, our goal is for the scoring program to assign the same scores as assigned to responses by the human raters. It is possible that the scores assigned by the human raters do not represent a correct response (or at least two or more human raters do not agree on the same score for the response). We will not consider the question of whether there is some difference between the scores assigned by human raters and the correct score for a response. This investigation is preliminary and a generally accepted benchmark is that any automatic scoring program should be able to produce a set of scores acceptable to human raters.

The prototype scoring program described in this paper fulfills the majority of these requirements. It is relatively easy to set up for an item. A content expert classifies the components of a response. The procedure has the potential to score on multiple scales (correct/incorrect, 0 - 5). The scoring procedure can learn from its mistakes and recognize a larger number of responses as it scores more and more responses. In a large number of cases, the score produced by the procedure is acceptable to human scorers.

Our scoring procedure is based on a pattern-matching approach. We differ from the procedure used by Carlson and Ward (1988) in the way our patterns are specified. Instead of attempting to represent all possible patterns, we seek a compact, high-level description of the patterns in the form of a grammar. This grammar and an associated dictionary can represent a large number of response patterns and can also be used as the basis for a procedure for scoring responses.

A Prototype Scoring Program

Approach

One fundamental problem with most pattern-based approaches to natural language computer processing is that all patterns that need to be recognized must be explicitly specified. In scoring a natural language response, this means that we would need to have, a priori all possible responses we wish to recognize. Each possibly recognizable response is called a pattern. If a pattern is missing, the responses matching this pattern will not be recognized. Naturally, the inclusion of all possible patterns that represent correct responses is critical to any scoring process.

Our approach to scoring involves using a sample of responses that are representative of all of the possible responses for an item. The scoring procedure uses the sample to construct a scoring key that takes the form of a grammar describing the "language" of responses. All possible patterns need not be specified because the grammar is a compact representation of the set of patterns.

The grammar consists of a set of rules for specifying the language of the set of responses. The grammar can be used to write sentences in a language and also to verify that a sentence is part of the language specified by the grammar.
In order to construct a grammar for the responses for a particular item, it is necessary to develop a set of semantic classes that represent the possible elements of a response. A response element is some portion (one or more words) of a response. We call these semantic classes because they partition the set of possible response elements into classes of meaning pertaining to the response. The semantic or meaning classes are used to classify the representative sample of responses. We will call this sample the response training set.

An example will help clarify the scoring procedure. The following excerpt and associated item were taken from a sample reading comprehension test developed and used in a study by Ward, Dupree, and Carlson (1987).

Geronimo was not the only grandchild of the Great Chief who confronted the Mexicans in hostile times. His cousin Nah-thle-tla, my mother, also encountered them during this period.

Nah-thle-tla was twelve years old when Haley's Comet appeared. She remembered "the night the stars fell." When she died at the age of one hundred and twelve, she was honored as the oldest woman in the United States. I inherited this quality from her. She also taught me all my earliest lore as my father, Nah-thle-tla's second husband, was killed in a buffalo stampede in my infancy.

We will consider the following item associated with this passage.

What characteristics does the author say he inherited from his mother?

The answer to this item is contained in paragraph two of the excerpt. The relevant sentences and phrases have been underlined. Some of the actual examinee responses given for this item are phrases like "long life", "longevity", "old age", "old agedness", "her old age characteristic", and "living to an old age." Each of these responses was considered a correct response for the item (scored 5 on a scale from 0 to 5). In the context of creating semantic classes, if we consider this sample of responses as representative of the kind of responses given for this item, we might initially develop two classes: length-phrase and life-phrase. The set of words or phrases belonging to the length-phrase semantic class would designate the length of life in the response. For the words and phrases in the sample responses, the words long, and old would belong to this semantic class. The life-phrase semantic class consists of words or phrases referring to life, living, or characteristics of being alive, such as age. The words live, living, age, and agedness would be part of this semantic class.

When all of the relevant words and phrases are classified into their respective semantic classes, a simple grammar can be constructed. The first part of the process to construct the grammar is to assemble strings of classifications for each of the responses in the training set of responses. The classification strings are formed by replacing each word or phrase in a response with its corresponding semantic class. For example, if the response was OLD AGE and OLD is part of the length-phrase semantic class and AGE is part of the life-phrase semantic class, the classification string formed is
One classification string is produced for each response of the response training set. Other possible classification strings are shown are

\[ \text{<length-life-phrase>} \text{ and } \text{<life-phrase>} \text{<length-phrase>} \text{<life-phrase>}. \]

Grammar rules for the language of responses can be constructed using classification strings. With the responses shown above we can construct the grammar rules shown in Table 1.

Table 1 - Response Grammar

\[
\begin{align*}
\text{(R1)} & \text{ <response-sentence>} \rightarrow \text{<length-phrase>} \text{<life-phrase>} \\
\text{(R2)} & \text{ <response-sentence>} \rightarrow \text{<life-phrase>} \text{<length-phrase>} \text{<life-phrase>} \\
\text{(R3)} & \text{ <response-sentence>} \rightarrow \text{<length-life-phrase>} \\
\text{(R4)} & \text{ <length-phrase>} \rightarrow \text{long} \\
\text{(R5)} & \text{ <length-phrase>} \rightarrow \text{old} \\
\text{(R6)} & \text{ <life-phrase>} \rightarrow \text{life} \\
\text{(R7)} & \text{ <life-phrase>} \rightarrow \text{age} \\
\text{(R8)} & \text{ <life-phrase>} \rightarrow \text{living} \\
\text{(R9)} & \text{ <length-life-phrase>} \rightarrow \text{longevity}
\end{align*}
\]

As seen in Figure 1, a sentence in a grammar is derived by using the rules of the grammar to rewrite or replace symbols enclosed in brackets (\(<\)>) with symbols that appear in the sentences of the language.

Figure 1 - Grammar Derivation for the Response Sentence

\[
\text{long life}
\]

\[
\begin{align*}
\text{(R1)} & \text{ <response-sentence>} \rightarrow \text{<length-phrase>} \text{<life-phrase>} \\
\text{(R4)} & \rightarrow \text{long <life-phrase>} \\
\text{(R6)} & \rightarrow \text{long life}
\end{align*}
\]

This grammar does not account for a response like "life that is long" because there is no rule that is part of the grammar that accounts for this particular ordering of words in a response. This is so because this response was not part of the training process. In the context of our scoring procedure, we would like the grammar to account for variations such as this without having to include individual responses in the training set.

This grammar can be used to generate sentences of this very restricted language. Phrases like old age, long life, and living long life can all be generated.

The production of a sentence of a language using a grammar is called a derivation. An example of a derivation of a sentence is depicted in Figure 1.
If a response is correct with respect to the item, it should be part of the language defined by the grammar of the response training set (given that the training set is sufficiently representative of the range of responses). Therefore, we should be able to find a derivation for it. Likewise, if the response is not correct, we should not be able to find a derivation for it.

Summarizing, the scoring process based on semantic pattern matching first involves a classification of the vocabulary of a representative set of responses into semantic (meaning) classes. The set of classified words and phrases is used to construct classification strings (strings of the semantic classes derived from responses). These classification strings are, in turn, used to construct a grammar that is subsequently used as the basis for a recognition procedure for correct responses from an item.

The next section describes a prototype semantic pattern-based scoring (SPAM-SCOR) program. This program implements the scoring protocol discussed in this section.

Program Description

The SPAM-SCOR program is a prototype that implements the methodology described in the previous section. The prototype scoring program can score a response as correct or incorrect. It also allows various scoring heuristics to be applied during the scoring process. For example, a response might include response elements that do not contribute to the correctness or incorrectness of the response. If these response elements were eliminated, the meaning of the response would not be changed. One scoring heuristic eliminates these extraneous response elements from responses.

There are certain capabilities the prototype scoring program does not have. The prototype cannot score on a scale and does not discriminate between answers that are incorrect and those that cannot be scored by the program.

SPAM-SCOR consists of three phases. In the first phase, the vocabulary of the response training set is classified into semantic classes. The second phase is responsible for creating the scoring key from a response training set. It is in this phase that the grammar for the language of correct responses is created. The third phase takes a set of responses for an item and scores those responses according to the scoring key prepared during the second phase.

In the first phase, a vocabulary list or lexicon is constructed from a training set of responses. The lexicon is assembled by forming a list of all the unique words appearing in the training set of responses.

Once the lexicon is assembled, the user specifies the number of semantic classes and the name of each semantic class needed to distinguish the words or phrases in the lexicon.

One training set response at a time is presented to the user. The user can classify one or more response elements by selecting the element/elements and assigning a semantic class to it/them. This is accomplished with a menu that is displayed consisting of a list of semantic classes. The menu also includes several auxiliary functions for manipulating the lexicon. A sample menu is shown below.
The next response is displayed when all the relevant response elements of the present response have been classified. The classification process continues until all responses of the training set have been considered. At the end of the process of classifying response elements, the lexicon will contain both the list of response elements and the semantic classification for each of the elements.

The lexicon and training set are used by the second phase of processing. The training set of responses undergoes several transformations before the grammar (scoring key) is created.

The first transformation that takes place is the removal or replacement of all the response elements that have been classified in the lexicon as response elements that can be removed or replaced. A standardized set of responses is produced from this transformation. Response elements classified as REMOVE in the lexicon were those that did not contribute to the meaning of a response. Their removal would not affect the meaning response. Replacements are made to standardize the vocabulary used in a response. Response elements having the same meaning or sharing the same root will automatically be replaced at this stage of processing. The process of removal and replacement is called canonicalization.

An example of canonicalization is shown in Figure 3. In this example, the response to be canonicalized is living to an old age. This response undergoes both kinds of transformations. The response elements to and an are classified as REMOVE in the lexicon. They are considered (for this item) not to add anything to the meaning of the response and consequently are removed in the canonicalized version of the response. The other transformation in this canonicalization is a replacement. In this example, the word living was replaced with the word live. Both live and living have the same meaning in the context of a response for this item.

Each canonicalized response is saved and sampled in the next part of processing. In the sampling process, responses are extracted to cover all lexicon entries. Any duplicate responses are removed at this time. An example of the sampling process will help clarify the process. Suppose the file of canonicalized responses is as shown in Figure 4.
In the sampling process, responses will be sorted according to the number of words in a response. All unique responses will be extracted and form the sampled set of canonicalized responses. The sampled set of responses for Figure 4 is shown in Figure 5.

Each response of the sample is converted into a classification string and saved. This file is used to create the scoring key by gathering all of the classification strings of the same length (number of classes) and removing any duplicate classification strings. The resulting list contains the possible semantic patterns that constitute a classifiable response. The scoring key is a dictionary of classification strings ordered by length.

The last phase of processing is scoring. There are two parts to the scoring process. In the first part, the file of item responses is canonicalized. The canonicalization is necessary because the responses of the training set were canonicalized. The second part of the scoring process is the scoring of each response.

A response is scored by transforming it into a classification string. In order for the response to be scored as correct, its classification string must be one of the classification strings in the dictionary of classification strings created from the response training set. If the classification string of the response is found in the dictionary, the response is correct. If the classification string is not found in the dictionary, the response is not correct. This process is the same as deriving responses with the grammar.

We conclude this section with some comments about the characteristics of the prototype scoring program. It is written in Common LISP and runs on a Sun SPARCStation I workstation. About 1200 lines of LISP code make up the training phase, the scoring key building
phase, and the scoring phase of the prototype program. On the SPARCStation I, the scoring of the sample set of data shown in Appendixes A and B required about 5 seconds. Canonicalization of the set of responses required 12 seconds of computing time. In its current form, the program could be easily ported to any computing platform supporting the Common LISP dialect. This would include both personal computers (using GOLD Hill Common LISP) and large IBM mainframes (using IBM Common LISP).

Preliminary Analyses of Program Functioning

Method

To achieve the second goal of this study, namely, to preliminarily compare the results assigned by the scoring program to those assigned by human raters, we carried out the following procedure on each of three sets of responses.

First, a sample of responses was selected from a set of responses for an item. This sample served as the training set for the whole set of responses. The criteria for selecting responses consisted of either selecting the first \( n \) unique responses, or just the first \( n \) responses.

Next, the selected responses were used to train the scoring program. A lexicon was formed, the elements of the lexicon were classified, and the dictionary of classification strings was created.

The set of training responses was machine scored to check that the scoring key was correct. If the training process was correct, the machine scores for these responses should be exactly the same as the human rater scores.

Following the check of the scoring key, the complete set of responses was scored using the dictionary of classification strings. The set of responses that were scored consisted of both the training set of responses and those responses that were not included in the training set. In the case of the third data set multiple scorings of the responses were carried out to investigate the effect of various scoring heuristics.

Two scoring heuristics were used. The first of these is called the ordering heuristic and the second is called the extraneous element heuristic. The ordering heuristic allows a classification string for a response to match whenever two classification strings are the same except for the order of their elements. The extraneous element heuristic allows two classification strings to match even if one contains extraneous or non relevant elements. These are ignored.

Following scoring, the machine scores were compared to the scores of the human raters. To obtain a more precise measurement of their agreement, we calculated \( \kappa \) to compare the human and machine scorings.

The first two data sets used in this evaluation came from the Ward, Dupree, and Carlson (1987) study that compared free-response and multiple choice items. The items consisted of a reading passage followed by alternating multiple-choice and free-response items. The \( n \) of the first data set was 78 and the \( n \) of the second data set was 40. The test was administered to volunteers from a state university. The sample was evenly divided between undergraduate juniors and seniors. About half were majors in the social sciences, with the remainder drawn
from the humanities, biological sciences, and natural sciences. The responses for the two selected items consisted of short answers of about three to five words.

The third set of data was derived from subjects who were using a microcomputer-based program of diagnostic assessment and instruction called GUIDES. The candidates of this sample of data consisted largely of community college students who were candidates for developmental reading programs. GUIDES consists of two separate programs. The program supplying our data focused on reading and study skills. The particular unit of this program from which data were derived was entitled "Understanding Text." The item selected for analysis in this study consisted of a reading passage and an item requesting the main idea of the passage.

The human rater scores for the data obtained from the Ward, Dupree, and Carlson (1987) study (data sets one and two) were the original scores for that study. The scoring of these data was carried out by a single support staff member under the direction of the researchers.

The GUIDES data were not originally scored. To create the training set for SPAM-SCOR, four ETS test development staff members volunteered to form a committee to score 150 randomly selected cases out of the total set of 635 responses. Each committee member independently scored the set of responses; then met as a whole to resolve any differences. The result was a training set of 150 responses whose scores were agreed upon by consensus. A set of notes was also compiled by the committee that described the process of scoring. These notes were helpful in understanding some of the nuances of the scoring decisions made by the committee and also provided some explanation for the discrepancies between the scores of the computer-based scoring procedure and the scores of the scoring committee.

To compare the machine scores of the third data set to a human rater's scores, we scored the remaining responses (those not part of the training set). This was done with a rubric created by committee. The rubric was produced as a result of scoring the 150 responses of the training set. Although the approach was less than optimal (we would have preferred a committee to score the all remaining responses), it provides some initial evaluation of the performance of the prototype scoring program.

Because of the complexity of the responses in the third data set, the process of scoring and analyzing these data was more iterative than in the case of the first two data sets. The process by which the third data set was analyzed is described below.

Prior to scoring the responses from the third set of data (we will call these data the GUIDES data for short), the responses were spelling-corrected with a commercial spelling correction program. An initial inspection of the GUIDES data showed a large number of spelling errors in this data set, which made spelling correction necessary. Incorrectly spelled words were located by the spelling corrector. A list of possible replacements was displayed and a replacement was selected. The spelling-corrected responses set was used for the training and scoring process. It should be noted that even after this process, a number of incorrect spellings remained in the response file. Although the spelling corrector recognizes words that are spelled incorrectly, it does not recognize incorrect spellings that accidentally form correctly spelled words.

The set of semantic classes created for the GUIDES data was based on the criteria established by the ETS committee. The initial set of semantic classes, which the committee considered necessary to form a minimal correct answer, consisted of a phrase denoting the
process of growing (the underlying process described in the passage), saltwater (the place where the growing was taking place), and food (the purpose for the growing).

In the case of the GUIDES data it was necessary to carry out the training process several times. Applying the first scoring key to the training set did not yield an accurate scoring of the training set. The set of semantic classes was fine-tuned each time the training process was done. By repeating the training process several times, a scoring key was produced that would result in scores for the response training set that were the same as those assigned to the training set by the scoring committee. The resulting set of semantic classes is shown below.

- future-growing-phrase
- future-denoting-phrase
- food-phrase
- growing-place-phrase
- growing-phrase
- growing-what-phrase
- grower-phrase

Once we were satisfied with the scoring key, we used the same scoring procedure used for data sets one and two on data set three. We were dissatisfied with the results of this scoring (71% accuracy) and decided to experiment with modifying the scoring procedure to enable it to recognize a larger percentage of responses.

The original scoring procedure uses the grammar that defines the language of correct responses without variation. If a grammar rule specifies that a response is to have a particular word or phrase order, this order is expected during the scoring process. In the variation to the original scoring program, the ordering heuristic and the extraneous element heuristic were applied. Both of these heuristics together allow for a greater class of correct responses to be recognized. The scoring process incorporating the two heuristics is called the relaxed scoring program.

Let us consider an example of how the heuristics are incorporated into the scoring program. For example, suppose a response grammar rule specified the following sequence of elements for a response:

\[
<\text{growing-phrase}> <\text{food-phrase}> <\text{growing-place-phrase}>
\]

A response grammar rule like this corresponds to the following response:

"The main idea of this article was to grow food in saltwater."

It is possible that the following variations of this response rule are also valid:

\[
<\text{food-phrase}> <\text{growing-phrase}> <\text{growing-place-phrase}>
<\text{growing-place-phrase}> <\text{growing-phrase}> <\text{food-phrase}>
\]

These classification strings may correspond to responses such as:

"The main idea of this article is to make food grow in saltwater."

and
"This article is about using saltwater for growing food."

By relaxing the strict interpretation of the order of elements in a response, the latter responses would be recognized as correct answers by the scoring procedure. Some variations of this response rule might result in a response that is not valid, for example,

"The main idea of this article is about growing saltwater in food."

We can relax the strict interpretation, but care must be taken that such relaxations do not cause invalid responses like the one above to be classified as correct.

Results

The composition of each of the data sets used for the preliminary evaluation of the prototype scoring program is shown in Table 2.

The first two data sets consisted of a small set of responses with an average length of response of 4 words. The third data set, supplied by GUIDES, consisted of a much larger set of responses with an average response length of 12 words. The responses of this set of data were for the most part complete sentences, as compared to the responses of the first two data sets, which were largely short phrases.

Table 2 - Scored Data Set Characteristics

<table>
<thead>
<tr>
<th>Data Set</th>
<th>N Scored</th>
<th>N Machine</th>
<th>Average Length Response in Words</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>79</td>
<td>25</td>
<td>3</td>
<td>short phrase</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>15</td>
<td>5</td>
<td>short phrase</td>
</tr>
<tr>
<td>3</td>
<td>635</td>
<td>485</td>
<td>12</td>
<td>sentence</td>
</tr>
</tbody>
</table>

Table 3 shows, for each of the data sets, the size of the training set, the size of the canonicalized training set, and a measure of the accuracy of the procedure that represents the correspondence between scores assigned by the scoring procedure and scores assigned by a human rater. An accuracy of 1.0 means that the scores assigned by human raters and those assigned by the machine were the same.
Table 3 - Comparison of Training Set Size and Performance Measure

<table>
<thead>
<tr>
<th>Data Set</th>
<th>N</th>
<th>Training Set N</th>
<th>Canonicalized N</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Training Set N</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>79</td>
<td>53(^1)</td>
<td>11(14%)</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>25(^1)</td>
<td>19(48%)</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>635</td>
<td>150</td>
<td>64(10%)</td>
<td>0.71(^2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.88(^3)</td>
</tr>
</tbody>
</table>

For the first two sets of data, there was no difference between the scores assigned by human scorers and those assigned by the prototype program. For the third set of data, only a 7 in 10 agreement was achieved with the original method of scoring and a 9 in 10 agreement was achieved by using the relaxed scoring procedure.

Charts 1 - 4 compare the number of scoring agreements to the number of scoring disagreements. Charts 1 and 2 show the scoring comparisons for data set one and two, respectively. The remaining charts show the agreements and disagreements for data set 3. Chart 3 shows the comparison for the original scoring procedure and Chart 4 represents the results obtained with the relaxed scoring procedure.

Chart 1 - Comparison of Scoring Agreement for Data Set 1

<table>
<thead>
<tr>
<th>Human</th>
<th>Machine</th>
<th>Right</th>
<th>Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Right</td>
<td>53</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Wrong</td>
<td>0</td>
<td>26</td>
</tr>
</tbody>
</table>

Chart 2 - Comparison of Scoring Agreement for Data Set 2

<table>
<thead>
<tr>
<th>Human</th>
<th>Machine</th>
<th>Right</th>
<th>Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Right</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Wrong</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Chart 3 - Comparison of Scoring Agreement for Data Set 3, Original Scoring Procedure (no heuristics)

1 Because the total number of responses was small and because the first two samples of data were primarily used to test the feasibility of the methodology, the total set of correct responses was used as the training set. In a more rigorous study, this would not be the case.

2 Method 1

3 Method 3
Chart 4 - Comparison of Scoring Agreement for Data Set 3, Relaxed Scoring Procedure (with heuristics)

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Right</td>
<td>Wrong</td>
</tr>
<tr>
<td>Right</td>
<td>105</td>
<td>3(0.6%)</td>
</tr>
<tr>
<td>Wrong</td>
<td>136(28%)</td>
<td>241</td>
</tr>
</tbody>
</table>

The final table of this section shows kappa for the two scoring methods used to score data set 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>$P_c$</th>
<th>Measure of Agreement</th>
<th>$P &lt; \kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.71</td>
<td>.43</td>
<td>.001</td>
</tr>
<tr>
<td>2</td>
<td>.88</td>
<td>.76</td>
<td>.001</td>
</tr>
</tbody>
</table>

Table 4 - Measure of Agreement Between Human Scoring and Machine Scoring of Response

Discussion

The data sets drawn from the Ward, Dupree and Carlson (1987) study supplied the original data used to test and debug the prototype scoring program. As can be seen from Charts 1 and 2, the results obtained match those of the human rater for all responses for both data sets.

It would be difficult, on the basis of these data sets, to conclude that the scoring methodology embodied in SPAM-SCOR would perform equally well on any set of responses that had the same characteristics, namely short responses on the order of three to five words and small $n$ ($n = 79$ for data set 1, $n = 40$ for data set 2). We do not know, for instance, how the program would perform given a set of responses of, say, 100 times the size of these data sets, and a training set of, say 1%, of the responses. This remains to be evaluated.

What we can say about the procedure, based on the first two sets of data, is that given a set of short responses and a representative training set, the scoring procedure works extremely well. We have confirmed with these data that (1) we can score natural language responses by inducing a grammar from a sample set of responses, (2) the resulting scores match those of a human rater given an appropriately representative set of responses, and (3) the resulting scoring key for this pattern-based approach is much smaller than what would be required by a traditional pattern-based approach where one would have to specify all possible patterns (14% for canonicalized data set 1 and 48% for canonicalized data set 2).
It is typical of pattern-based approaches to natural language processing that they begin to exhibit problems as the complexity of the patterns to be recognized increases. If a pattern is missing from those that are to be recognized, the pattern matcher will miss it when it is encountered in a set of responses. This failing of pattern-based approaches was the main reason they were left behind for more sophisticated approaches. The question that remains, however, is what can be done to a pattern-based approach to increase its ability to recognize valid patterns. The third data set is of sufficient complexity so that this fundamental problem with pattern-based approaches begins to arise. Because of this, we were able to explore variations in the basic pattern-based approach.

In scoring the GUIDES data we encountered several problems not encountered in analyzing the data from the Ward, Dupree, and Carlson study (1987). These were (1) the scoring program was not able to correctly score the training set after a single training, (2) the scoring program produced poor results (only 71% accuracy), (3) some responses were scored correct when in fact they were incorrect and vice versa, and (4) subtle differences in responses were difficult to recognize. We will consider each of these problems individually.

(1) The scoring program was not able to score the training set after a single training.

Several iterations of training were necessary because certain subtleties of responses had to be captured during the training process if the scoring procedure was to score the training set correctly. In the first attempt at creating a proper scoring key, several responses of the initial 150 responses were scored incorrectly. After several attempts at training to eliminate any incorrectly scored training set responses, we arrived at a proper set of semantic classes.

For complex responses, the set of semantic classes must be carefully constructed. If this is not the case, one or more problems may arise. First, it may be that some important distinction is not captured in the semantic classification. One of the responses that was problematic for the scoring program and the scoring committee was "Food that can be grown on saltwater." Although this response contains the main elements, it does not suggest the experimental or future nature of the process being described by the passage (according to this scoring committee). Phrases like this made it necessary to distinguish between those phrases that designate growing in the present (or past) versus those designating growing in the future. This is the reason for the classes future-growing-phrase and future-denoting-phrase, which both connote an action occurring in the future.

Another problem that may arise during the formation of the semantic classes has to do with what we will call semantic class conflict. At certain times during the classification process we found that we wanted to classify the same word or phrase in more than one way. This of course created a problem, because an underlying assumption of the present scoring procedure is that only one class will be assigned to a response element.

(2) The scoring program produced poor results for the GUIDES data (71% accuracy).

Given the excellent performance with the first two sets of data, this result for the GUIDES data was surprising. One reason for the scoring program to incorrectly score a response is a lack of an appropriate classification string that matches the classification string for the response. This means that the training set had no response that resulted in the creation of an appropriate classification string. A response might be correct except that it uses a synonym for one of the words that would be recognized. A response might be correct except that it has some extra...
words or phrases that do change the meaning of the response. The scoring program used to score the first two data sets, and initially the third data set, did not account for these possibilities. The first of these problems is addressed in the alternative scoring program by relaxing the strict ordering requirement for classification strings and by ignoring extraneous words or phrases in a response. Making these changes resulted in better performance. Other scoring heuristics may also improve program performance.

The latter problem, that of encountering a word or phrase that is unknown to the scoring program but represents a correct element in the response, could be dealt with by supplying a list of synonyms.

(3) Some responses were scored correct when in fact they were incorrect and vice versa.

Both scoring procedures produced scores that disagreed with those of the human raters. Two kinds of disagreements were exhibited. The first were those where the rater scored the response correct and the program scored the response incorrect. We will call this a D-disagreement. In the second kind of disagreement, the rater scored the response incorrect and the computer scored the response correct. This type of disagreement will be called a U-disagreement. In the case of the original scoring program, the program scorings erred on the side of many more D-disagreements. The relaxed scoring program had fewer disagreements overall, but a larger number of U-disagreements.

The original scoring program produces more D-disagreements than the relaxed scoring procedure because it depends more on the correct pattern being present in the scoring key in order to recognize a correct response. Because it cannot use any general rules to manipulate the scoring key (as in the relaxed scoring procedure), its ability to recognize a greater range than what is specified by the training set is limited. Likewise, the original scoring program would naturally have much better performance in the area of U-disagreements. A U-disagreement will occur when a response does result in a classification string that is included in the scoring key, but there is some subtlety that is part of the response that is not accounted for in the classification strings making up the scoring key. An example of a response of this type is *Food that can be grown on saltwater.* This response has all of the necessary elements for a correct response (growing phrase(grown), location phrase(saltwater), and a food phrase(food)) but is still considered incorrect by the scoring committee. The committee indicated that the phrase *Food that* would be a tip-off for an incorrect answer. Unfortunately this feature was not consistent throughout the set of responses (Fiero, personal memorandum, August 17,1989).

The relaxed scoring program yielded the opposite performance, albeit overall somewhat better, than the original scoring program. The number of D-disagreements for the relaxed procedure was roughly half the number of the original scoring program. On the other hand, the number of U-disagreements for the relaxed scoring program was four times more than produced by the original program.

If we relax the constraint of order of necessary elements, it may be the case that a response to be scored will violate some requirement for a correct answer that strict ordering maintains. Therefore, some orderings of response elements may in fact be incorrect responses, but because order is no longer considered by the relaxed scoring program, these responses will be scored correct.
The reduction in the number of D-disagreements is also attributed to the relaxation of the ordering constraint. If we can find the elements of a classification string in a response, regardless of the order they are in, we possibly may have recognized a correct response. On the whole, the relaxed scoring procedure yields an overall performance that is much better than the strict scoring procedure (88% as opposed to 71%).

(4) Subtle differences in responses are difficult to recognize using a pattern-based approach.

This problem is perhaps the most difficult to deal with when using pattern-matching approaches. This has to do with subtle semantic differences in responses, as was shown by the example "Food that can be grown in saltwater." Responses such as this would generally have to be dealt with on a case-by-case basis.

As noted in the introduction, one of the criteria for an automatic scoring program should be that it is accurately able to identify those responses that it cannot score. In the analysis of these results we see that for complex responses (12 to 15 words), the original scoring procedure produces a very low number of U-disagreements and a very high number of D-disagreements. If the number of U-disagreements were zero, all of the responses scored wrong by the scoring procedure could be reviewed by a human rater. Based on the results for the GUIDES data, 191 responses were scored as correct. This accounts for 22% of the total number of responses scored, leaving 78% to be reviewed. On the other hand, the relaxed scoring program scored 210 responses correct (43%), but 14 (3%) of these were incorrectly scored. Although the relaxed procedure recognizes many more responses, it does not facilitate a separation of responses into those that are correct and those that need to be reviewed as well as the original scoring procedure does. Neither scoring program delineates the set of responses into disjoint sets.

An important aspect of any automated scoring procedure is how well it agrees with human raters' scoring. We computed $\kappa$ for the two scoring programs (original and relaxed); the results are shown in Table 4. According to Landis and Koch (in Fleiss, 1981, p. 216), a value of the agreement measure greater than .75 represents excellent agreement beyond chance. Between .40 and .75 represents fair to good agreement. For the original scoring program our agreement is at the low end of this range and the relaxed scoring program falls at the high end of the range. What this may tell us is that even though the relaxed scoring program had a larger number of U-disagreements, overall its scores corresponded very well to a human rater's scores.

One final comment should be made about the performance of SPAM-SCOR as compared to the study of Carlson and Ward (1988). In the present study, a pattern-matching approach to scoring yielded 55% of the responses correctly scored. After the program was revised, a second set of responses was scored, showed 46% of them scored correctly. In the first scoring of the GUIDES data, 71% of the responses were scored correctly. With the revised program, 88% of the responses were scored correctly. We would have expected this result since the scoring key constructed by SPAM-SCOR should account for many more correct patterns than an approach that individually specifies the correct patterns.

**Summary and Conclusions**

In this paper we have described an automatic procedure for scoring natural language free-response items. The procedure is pattern based and is trained on a sample set of responses to create the scoring key. A prototype program called SPAM-SCOR has been written
to implement the scoring procedure. The program was used to analyze three sets of data. Two of these sets of data consisted of short responses (3 to 5 words). The third set of data consisted of longer responses (12 to 15 words).

When the responses were short and the training set relatively representative, the scoring program matched the scoring of a human rater without disagreement. For longer responses, the training and scoring process was more complex because of the complexity of the responses, with the result of poorer performance of the scoring program.

As noted at the beginning of this report, one of the goals for this research was to investigate the feasibility of alternative procedures for scoring natural language response items. We have shown a procedure that is relatively inexpensive to set up (no large knowledge bases required) and that, under certain circumstances, produces response scores that agree with those of human raters. But it is still an open question whether this procedure and its associated program represent a viable approach to scoring natural language responses. The question of whether to proceed is based on two considerations. First, is there interest in using free-response natural language response items on examinations? And second, can the problems known about the process at this stage of development be addressed in such a way as to improve the performance of the scoring procedure?

Our answer to the first of these questions is that yes, there is interest in these items and therefore it would seem that finding some way to automatically score them would be a useful activity. The purpose of this paper, though, is not to argue the merits of these alternative kinds of items, but rather, because interest does exist, to extend the state of the art of scoring them.

Given the interest in these items, it would make sense to pursue the development of this scoring methodology. Because the scoring methodology shows promise (at least for short responses), the development of the procedure may produce a viable alternative to scoring these items manually.

We indicated that one of the problems arising while scoring the GUIDES data was the presence of response elements having the same meaning as those that are part of the lexicon but were not classified in the training process. When a correct response was encountered consisting of words or phrases not classified, it was incorrectly scored as incorrect by the scoring program.

A remedy for this problem would be to incorporate a thesaurus into the scoring program. When a response element is encountered that is unknown to SPAM-SCOR, a search for this response element can be made in the thesaurus. If the element is a synonym for some response element that is part of the SPAM-SCOR lexicon, the unknown element can be classified as belonging to the same semantic class as the response element in the SPAM-SCOR lexicon. Including a thesaurus into SPAM-SCOR would increase the number of recognizable responses.

The names of the semantic classes can be used in a similar way. We may not be able to identify the membership of a response element strictly by looking to see if it is a synonym for some element that is part of a particular semantic class. In this case, the identification may be possible by examining the synonyms that correspond to the name of the semantic class. Some standard naming of the semantic classes would be necessary for the name of the class to be used in this way.
Dictionary definitions could be used to determine if a response element belongs to a particular semantic class. Of course, this is a significantly more complex application. It would involve associating dictionary definitions in such a way as to know how they are related. Optimally this could be carried out automatically. The technology to do this is quite sophisticated and still in the formative stages of research. Another approach would be to hand-encode the dictionary definitions in such a way that their relationship was apparent from this encoding. The encoding process could be quite complex. It might be possible to obtain this encoding "off-the-shelf" in an existing product. This is an area to be explored.

An important problem that arose out of the analysis of the GUIDES data was the introduction of responses that were scored correct by the machine scoring procedure but incorrect by the human rater. The occurrence of such incorrectly scored responses makes it impossible to consider examining only one set of responses once they have been scored by the scoring program. We would like the scoring program to be able to distinguish between those it can score and those it cannot. All those that cannot be scored would be flagged for a human rater.

Part of the problem of having more responses scored correct that are incorrect (according to a comparison with a human rater's scores) has to do with the relaxation of the ordering constraint for the grammar. This was done to make the classification string derived from the response training set account for a greater number of responses. The problem with doing this, though, is that some of the responses that are scored correct may be incorrect as a result of the order of the response elements. A more rigorous approach to the specification of patterns would seem necessary.

One possible approach is to incorporate a parser into the scoring program. When a response is processed by the scoring program it will first be parsed into some appropriate representation. This representation becomes the classification string for the response. More than one response may be parsed into the same representation so the description of patterns still remains smaller than what would be required in a strictly pattern-based approach. By incorporating a syntactic aspect to the process of training and scoring we can maintain important syntactic relationships that have bearing on whether a response is correct or incorrect. The variation in possible patterns is accounted for by the grammar that produces the parse. The problems associated with completely relaxing the order of elements in a response is avoided as the syntax constrains the possible orderings.

We also would like the scoring program to score on different scales. For example, we would like to be able to score responses on a scale from 0 to 5.

There are two possible approaches to accomplish this using the present scoring procedure. In the first, the scoring algorithm is extended to be able to be trained with multiple responses. There will be a training set for responses assigned a 0 score, one for responses assigned a score of 1, and so on for each possible score. Each scoring key from each training set is then used to score a response. The key to which the response comes closest determines the score of the response. This would require the development of a metric for determining "closeness" to a scoring key. One possible way to implement this metric is to assign a weight value to the various elements of a response. Each scoring key (for 0, 1, etc.) contains some of the elements of a correct answer. A score is constructed for each of the keys based on the weight values. The response is scored by computing the total weight for its scorable elements, and this score is compared to each of the scoring keys. The scoring key score having the least
difference in comparison to the score of the response is used to determine the final score of the response.

An alternative to this approach would be to use a single training set consisting of correct responses. This training set gives the necessary semantic classes for scoring. Each of the semantic classes is assigned a weight according to its importance with respect to the item. When a response is scored, if it does not have every necessary semantic class, a score is calculated for that response by subtracting from a perfect score the weights of the missing classes. This method differs from the first method in two ways. First, only one training set is required. Second, the features recognized in a response are specified in the semantic classes of the training set for the correct response whereas in the first method, the features that discriminated between responses were implicit in each of the training sets.

It is our belief that making these modifications to the prototype scoring program will increase the performance of the program substantially. The pattern-based approach to scoring offers an alternative to more expensive natural language processing techniques. The approach seems most appropriate since we are trying, in the scoring process, to recognize correct (or some degree of correct) responses. The process of recognition is well suited to pattern-based approaches.

The question remains as to what the benefit of this approach will be if a viable scoring system can be built based on this methodology. Several benefits can be anticipated.

Of course, as we have said previously, an automatic scoring procedure may be substantially more cost effective than manual scoring of these types of responses. Initially, the construction of the sample set of responses and semantic classes may be costly, but once this is done, as long as the item is used the costs for scoring the item are only incremental as the key is modified to accept new variations in response language.

The second benefit to be obtained by using this procedure is that the scoring of these items will be more consistent. As long as the same key is used, the score assigned to a single response over multiple occurrences of that response will be the same. The scoring of the GUIDES data by the ETS scoring committee demonstrated that different human raters will often score a response differently.

A third benefit to pursuing investigation is that it surfaces a number of important issues if we are going to use constructed response items on examinations. Given that it is possible to score the items, there are still many psychometric issues to be resolved: do we have a scale on which to interpret item scores, can we equate items of this kind across multiple forms, and can we assess how an item might operate differentially? For many constructed-response items, issues like these have had little investigation. This scoring procedure affords us an opportunity to look at one kind of constructed-response item to explore these issues.

The fourth benefit of this procedure is that there is a possibility that the scoring procedure can be generalized to other domains. A scoring program such as this could also be used for certain kinds of mathematics items. The patterns need not be sequences of words. In a future evaluation of SPAM-SCOR we will analyze several different kinds of mathematical items to determine whether the procedures can be generalized.
For these reasons we see the development of SPAM-SCOR as both important and beneficial. Constructed-response items are going to be a part of our future examinations, and SPAM-SCOR represents one possible means of scoring this new type of item.
Bibliography


APPENDIX A

Transcript of Item 54
(Longevity Item)
Transcript of the Training Process

> (stage-one "54")

.. lexicon report..

.. the total number of words read were 144..
.. the total number of words in the lexicon is 27..
.. the percentage of lexicon of the total words is 18.75..

Define Semantic Classes
How many classes? 3

Enter name of class 1: life-phrase
Enter name of class 2: length-phrase
Enter name of class 3: life-length-phrase  } semantic class specification

Pattern Classification
classify: (LONGEVITY)  } training dialogue
Word or Phrase Context: (LONGEVITY)

Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT
:3
96.29% of vocabulary that remains to be classified.

Canonicalization: (LONGEVITY)
Semantic Pattern for Sentence: (LIFE-LENGTH-PHRASE)

Ok (y=yes, n=no): y
Sentences Remaining To Be Classified: 21

classify: (HARDINESS)
Word or Phrase Context: (HARDINESS)
Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT

92.59% of vocabulary that remains to be classified.

Canonicalization: (HARDINESS)
Semantic Pattern for Sentence: (LIFE-LENGTH-PHRASE)

Ok (y=yes, n=no): y
Sentences Remaining To Be Classified: 20

(OLD AGE)
1 2
N N

First element of pattern(n if finished, x to abort): 1
Second element of pattern: 1
classify: (OLD)
Word or Phrase Context: (OLD AGE)

Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT

88.88% of vocabulary that remains to be classified.

(OLD AGE)
1 2
Y N

First element of pattern(n if finished, x to abort): 2
Second element of pattern: 2
classify: (AGE)
Word or Phrase Context: (OLD AGE)
Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT
:1
85.18% of vocabulary that remains to be classified.

Canonicalization: (OLD AGE)
Semantic Pattern for Sentence: (LENGTH-PHRASE LIFE-PHRASE)

Ok (y=yes, n=no): y
Sentences Remaining To Be Classified: 19

(LONG LIFE)
1 2
N N

First element of pattern(n if finished, x to abort): 1
Second element of pattern: 1
classify: (LONG)
Word or Phrase Context: (LONG LIFE)

Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT
:2
81.48% of vocabulary that remains to be classified.
(LONG LIFE)
  1 2
  Y N

First element of pattern(n if finished, x to abort): 2
Second element of pattern: 2
classify: (LIFE)
Word or Phrase Context: (LONG LIFE)

Select a class from the following list of classes:
  1 LIFE-PHRASE
  2 LENGTH-PHRASE
  3 LIFE-LENGTH-PHRASE
  4 REPLACE
  5 MODIFY
  6 REMOVE
  7 SHOW
  8 UPDATE-CLASSES
  9 ABORT
 :1
77.77% of vocabulary that remains to be classified.

Canonicalization: (LONG LIFE)
Semantic Pattern for Sentence: (LENGTH-PHRASE LIFE-PHRASE)

Ok (y=yes, n=no): y
Sentences Remaining To Be Classified: 18

(ABLE TO LIVE A LONG TIME)
  1 2 3 4 5 6
  N N N N Y N

First element of pattern(n if finished, x to abort): 3
Second element of pattern: 3
classify: (LIVE)
Word or Phrase Context: (ABLE TO LIVE A LONG TIME)

Select a class from the following list of classes:
  1 LIFE-PHRASE
  2 LENGTH-PHRASE
  3 LIFE-LENGTH-PHRASE
  4 REPLACE
  5 MODIFY
  6 REMOVE
  7 SHOW
  8 UPDATE-CLASSES
  9 ABORT
 :1
74.07% of vocabulary that remains to be classified.

(ABLE TO LIVE A LONG TIME)
1 2 3 4 5 6
N N Y N Y N

First element of pattern(n if finished, x to abort): 5
Second element of pattern: 6
classify: (LONG TIME)
Word or Phrase Context: (ABLE TO LIVE A LONG TIME)

Select a class from the following list of classes:
1 LIFE-PHRASE
2 LENGTH-PHRASE
3 LIFE-LENGTH-PHRASE
4 REPLACE
5 MODIFY
6 REMOVE
7 SHOW
8 UPDATE-CLASSES
9 ABORT

:2
70.37% of vocabulary that remains to be classified.

(ABLE TO LIVE A LONG TIME)
1 2 3 4 5 6
N N N N Y Y

First element of pattern(n if finished, x to abort): n

.. all of the words or phrases in this sentence have not been classified. Classify remaining elements as REMOVE? (y-yes,n-no): y
..classify ABLE class: REMOVE ..
..classify TO class: REMOVE ..
..classify A class: REMOVE ..

Canonicalization: (LIVE LONG TIME)
Semantic Pattern for Sentence: (LIFE-PHRASE LENGTH-PHRASE)

Ok (y=yes, n=no): y
Sentences Remaining To Be Classified: 17

... and this process continues for all responses of the training set ...

Transcript of the Construction of the Scoring Key

> (stage-two *54*)
.. restoring the lexicon..

.. restoring lexicon element LONGEVITY..
.. restoring lexicon element LONGEVITY..
.. restoring lexicon element LONGEVITY..
.. restoring lexicon element HARDINESS..
.. restoring lexicon element HARDINESS..
.. restoring lexicon element HARDINESS..
.. restoring lexicon element AGE..
.. restoring lexicon element AGE..
.. restoring lexicon element LONG..
.. restoring lexicon element LONG..
.. restoring lexicon element LONG..
.. restoring lexicon element LIFE..
.. restoring lexicon element LIFE..
.. restoring lexicon element LIFE..

... and so on...

.. canonicalizing the training file..

old sentence: (LONGEVITY)
new sentence: (LONGEVITY)
old sentence: (HARDINESS)
new sentence: (HARDINESS)
old sentence: (LONGEVITY)
new sentence: (LONGEVITY)
old sentence: (OLD AGE)
new sentence: (OLD AGE)
old sentence: (LONGEVITY)
new sentence: (LONGEVITY)
old sentence: (LONGEVITY)
new sentence: (LONGEVITY)
old sentence: (OLD AGE)
new sentence: (OLD AGE)
old sentence: (LONG LIFE)
new sentence: (LONG LIFE)
old sentence: (LONGEVITY)
new sentence: (LONGEVITY)
old sentence: (ABLE TO LIVE A LONG TIME)
new sentence: (LIVE LONG TIME)

... and so on...

elapsed time: 7 seconds
The training set is sampled

Classification strings are constructed

Gather responses and classification and strings
.. dictionary ..

((1 ((LIFE-LENGTH-PHRASE) (LENGTH-PHRASE)))  
(2 ((LENGTH-PHRASE LIFE-PHRASE)  
  (LIFE-PHRASE LIFE-LENGTH-PHRASE)  
  (LIFE-LENGTH-PHRASE LIFE-PHRASE)  
  (LIFE-PHRASE LENGTH-PHRASE))))

(3 ((LENGTH-PHRASE LIFE-PHRASE LIFE-LENGTH-PHRASE))))

.. created the dictionary of classification strings

.. collected patterns ..
.. processing dictionary entries of length 1 ..
.. processing dictionary entries of length 2 ..
.. processing dictionary entries of length 3 ..

((1 (LIFE-LENGTH-PHRASE)  
  (LENGTH-PHRASE)
  ((LONGEVITY) (HARDINESS) (OLD)))
(2 (LENGTH-PHRASE LIFE-PHRASE)  
  (LIFE-PHRASE LIFE-LENGTH-PHRASE)  
  (LIFE-LENGTH-PHRASE LIFE-PHRASE)  
  (LIFE-PHRASE LENGTH-PHRASE)  
  ((OLD AGE) (LONG LIFE)  
    (AGE LONGEVITY)  
    (LONGEVITY LIFE)  
    (LIVE LONG)  
    (LIVE LONG TIME)  
    (LIVE OLD AGE)  
    (LIVE LONG LIFE)))
(3 (LENGTH-PHRASE LIFE-PHRASE LIFE-LENGTH-PHRASE)  
  ((OLD AGE LONGEVITY))))

.. stage two completed..

Transcript of Response Scoring

> (score "54" 'score3)

.. restoring the lexicon (if necessary) ..

.. canonicalizing responses ..

elapsed time: 12 seconds

.. scoring responses ..

elapsed time: 5 seconds

.. scores ..
(CORRECT LONGEVITY)
(CORRECT HARDINESS)
(CORRECT LONGEVITY)
(CORRECT HE IS OLD TOO)
(CORRECT OLD AGE)
(CORRECT LONGEVITY)
(CORRECT OLD AGE)
(CORRECT LONG LIFE)
(CORRECT LONGEVITY)
(INCORRECT THAT SHE WAS THE OLDEST WOMAN IN UNITED STATES)
(CORRECT ABLE TO LIVE A LONG TIME)
(CORRECT LIVING TO AN OLD AGE)
(CORRECT LONG LIFE)
(CORRECT BEING ABLE TO LIVE A LONG LIFE)
(CORRECT OLD)
(CORRECT LIVING UNTIL AN OLD AGE)
(INCORRECT OLDEST WOMAN IN UNITED STATES)
(CORRECT OLD AGE LONGEVITY)
(CORRECT OLD AGE VITALITY)
(CORRECT AGE LONGEVITY)
(CORRECT OLD AGEDNESS)
(INCORRECT HAVING A GREAT MEMORY)
(INCORRECT MEMORY)
(CORRECT LONGEVITY OF LIFE)
(CORRECT TO LIVE TO AN OLD AGE)
(INCORRECT A GREAT MEMORY)
(CORRECT OLD AGE)
(CORRECT LONGEVITY)
(CORRECT LONG LIFE)
(INCORRECT A SENSE OF PRIDE)
(CORRECT LONG LIFE)
(CORRECT OLD AGE)
(CORRECT OLD AGE)
(CORRECT OLD AGE)
(CORRECT OLD LIFE)
(CORRECT OLD LIFE)
(INCORRECT THE HONOR OF OLDEST WOMEN IN THE UNITED STATES)
(CORRECT THE QUALITY OF OLD AGE)
(CORRECT LIVING TO AN OLD AGE)
(CORRECT LONGEVITY)
(CORRECT HER OLD AGE CHARACTERISTIC)
(CORRECT LIVING TO BE A RIPE OLD AGE)
(INCORRECT THE COURAGE TO CONFRONT MEXICANS)
(INCORRECT OLDEST WOMAN IN UNITED STATES)
(INCORRECT SENTIMENTALITY)
(INCORRECT HORRORS)
(CORRECT LONG LIFE)
(CORRECT OLD AGE)
(CORRECT OLD AGE)
(CORRECT ABILITY TO LIVE A LONG LIFE)
(INCORRECT THE ART OF COOKING)
(CORRECT LONGEVITY)
(CORRECT LONGEVITY)
(CORRECT LONGEVITY)
(CORRECT LIVING TO AN OLD AGE)
(CORRECT LIVING LONG)
(INCORRECT HER GREAT MEMORY)
(CORRECT LONG LIFE)
(CORRECT OLD AGE)
(INCORRECT OLDEST WOMAN IN THE UNITED STATES)
(INCORRECT HONOR)
(CORRECT OLD AGE)
(CORRECT OLD AGE)
(INCORRECT OLDEST PERSON IN THE UNITED STATES)
(INCORRECT COURAGE STRENGTH AND LOVE OF LIFE)
(CORRECT THAT OF LONG LIFE)
(CORRECT THE ABILITY TO REMEMBER MANY THINGS OF LONG AGO)
(INCORRECT HER HONOR)
(INCORRECT THE WILLPOWER TO LIVE)
(CORRECT OLD AGE)
(INCORRECT HER DEXTERITY)
(INCORRECT BEING HONORED)
(CORRECT OLD AGE)
(INCORRECT PRIDE IN FAMILY AND APACHES)
(CORRECT OLD AGE)
(CORRECT OLD AGE)
(CORRECT LONGEVITY)
(INCORRECT BEING THE OLDEST WOMAN IN THE UNITED STATES)
(CORRECT LIVING TO A VERY OLD AGE)
(CORRECT LONG LIFE)
APPENDIX B

Pickleweed Item
Passage and Responses
Pickleweed, Palmer’s and Saltbush
Can we grow tomorrow's food in today's saltwater? The answer is: possibly. Millions of acres of seashore lie empty. Billions of gallons of ocean water are available for irrigation. The challenge is to find a way to use them to grow nourishing crops. But the plants humans use for food cannot be grown with saltwater. Scientists over the years have tried several approaches. Some have tried to remove the salt from the seawater so that the water could be used for irrigation. Others have tried to grow traditional crop plants, such as wheat and rice on coastal land, hoping to find a variety that could tolerate saltwater. These efforts have met with very little success.

A new approach seems more promising. Scientists at the Environmental Research Lab have explored saltwater marshes and swamps and they have identified an astonishing variety of plants that can live on saltwater. Such plants are called halophytes. The search has covered the United States and parts of Asia, Africa, Australia, and Latin America. The scientists have found several varieties of halophytes that are equal or even superior in nutritional value to wheat, rice, and alfalfa, commonly used as livestock feed. These findings have seemed promising enough from a commercial standpoint that a major food-processing firm has agreed to finance continuing research.

The growing of halophytes could become an important part of world agriculture. There are over 20,000 miles of desert coastline in the world, and another billion acres of desert land, about 10 times the area of California, lie in underground reservoirs of saltwater. With that much land and that much available saltwater for irrigation it is clear that millions of tons of food could be produced every year. The scientists' goal is to find the edible halophytes that are most suitable for production.

The staff at the Environmental Research Lab think they already have found a few. A halophyte known as Palmer's grass, that grows along the Gulf of California, produces a grain that was once used by the Cocopa Indians. The grain is rich in carbohydrates and could be used to make flour. A plant called pickleweed is already used in Europe as a salad ingredient and could find similar use in the United States. Still another promising plant is the saltbush. Its leaves contain from fifteen to twenty proteins, about the same number of proteins found in alfalfa. The high protein content suggests that the saltbush could be used a livestock food, and, in fact, it has been fed to goats on an experimental basis. An acre of land planted with saltbush yields twice as much food as an acre of alfalfa. If a way to remove the excess salt from the leaves could be perfected, the saltbush would probably be the halophyte most suitable for commercial growing.

Whether farmers could be persuaded to grow these plants is another question. But the scientists at the Environmental Research Lab are confident that they have found a way to increase the world's food supply for a growing world population.
APPENDIX C

Sample of Machine Scored Pickleweed Responses
Sample of Scored Responses

(INCORRECT THE GROWING OF HALOPHYTES)
(CORRECT CAN WE GROW TOMORROWS FOOD IN TODAYS SALTWATER QUESTION-MARK)
(INCORRECT EXPERIMENTING WITH FOOD AND SALTWATER FOR MORE PRODUCTION OF FOOD)
(INCORRECT THREE PLANTS THAT CAN SURVIVE IN SALTWATER ARE PICKLEWEED PALMERS GRASS AND SALTBUHSH)
(INCORRECT THE MAIN IDEA STATES THAT SCIENTISTS ARE EXPERIMENTING WITH PLANTS TO SEE IF THEY CAN BE GROWN IN WATER)
(INCORRECT SCIENTISTS ARE FINDING NEW WAYS TO PRODUCE FOOD IN THE WATER)
(CORRECT GROWING FOOD IN SALT WATER)
(INCORRECT THE CHALLENGE IS TO FIND A WAY TO USE THEM TO GROW NOURISHING CROPS)
(CORRECT WE CAN GROW FOOD IN SALTWATER)
(INCORRECT TRYING TO FIND A USE FOR DESERT LAND AND DESERT COASTLAND TO IMPROVE HUMAN USES)
(CORRECT THAT SALT WATER YOU CAN GROW FOOD)
(INCORRECT THE POSSIBILITY OF USING SALT WATER TO GROW PLANTS)
(CORRECT OCEAN WATER CAN POSSIBLY BE USED TO GROW CROPS)
(CORRECT WAYS THAT SALTWATER CAN BE USED IN GROWING FOOD FOR THE FUTURE)
(CORRECT GROWING FOOD THAT WE EAT TODAY OR THAT WE COULD EAT IN SALTWATER)
(CORRECT CAN WE GROW TOMORROWS FOOD IN TODAYS SALTWATER)