# Computer Monitored Problem Solving Dialogues

Lisa Dion, Jeremy Jank, Nicole Rutt

July 29, 2011

**Abstract**

This project "looks over the shoulder" at students collaboratively engaged in a math problem-solving activity. One task we looked at was mechanically classifying the students current activity or knowledge state, of which we have identified 15 different categories. We have produced an automatic classifier that examines student sentences and is 55% accurate in identifying utterances as containing certain bits of knowledge or evidence of certain activities. The classifier was built from a corpus of student-written reports. Treating each sentence as a bag of words, we built vector space models of the word co-occurrence matrix using both non-negative matrix factorization (NMF) and latent semantic analysis (LSA). Classification was achieved by comparing new, unknown, sentences with pre-built bundles of manually tagged sentences, one bundle for each classification.

Our categories are specific to the problem being solved, particular bits of knowledge needed to understand a two-person game called Poison. We have also been characterizing the dialogues with problem-independent categories: a math collaborative dimension and a problem-solving dimension. This will enable us to classify utterances with regard to in what ways students are participating in the dialogue and the problem-solving process. The context of this work is a quantitative problem-solving course in which students work in small groups. Our goal is for the computer to notice some of the same aspects of the activity that a teacher walking around the classroom might observe, such as what realizations a group has achieved and how students are collaborating. This type of computer-mediated collaborative problem solving exposes student thinking, providing opportunities to gain insights about student learning.

## I. Introduction

The classification of student understanding and participation upon the analysis of dialogues has been the emphasis of our project this summer. The dialogues come from a collaborative math problem-solving environment, a classroom setting where groups of students (3-4 students per group) worked together on a certain math problem-solving activity. Group reflection reports which students wrote after completing the activity as well as transcripts of tape recordings of some groups playing the problem-solving game in real-time, were the data we wanted to classify. In particular, the data under examination were sentences within the reports and transcripts.

In order to achieve the task of classifying the dialogues, first we had to manually tag each utterance within the reports and transcripts as exemplifying a particular bit of realization or participation within either the specific problem-solving activity or the more general collaborative math environment. Then, we built classifiers using different methods in order to automatically categorize sentences, since our objective this summer was to construct a classifier that accurately identifies student utterances into certain categories, in regards to the human classification of utterances. Our automatic classifier is 55% accurate in placing student sentences into particular classes, in which there are 15 of these domain-dependent problem-solving activity classes. Furthermore, the classifier was able to correctly identify 96% of the sentences, which were human-classified as representing a distinct activity, and 63% of the sentences, which were classified as containing a certain bit of knowledge.

Currently, students in the Quantitative Problem Solving class collaborate as groups, around a table in the classroom, while participating in a problem-solving activity. Ultimately though, the goal is to move at least some of the problem-solving activities of the Math 110 class online. This online setting will be a computer-monitored collaborative environment where students can chat in real-time with their partners all the while taking part in the problem-solving process. This chat-room environment allows for student conversations and problem-solving actions taken by the students to be recorded and monitored.

The dialogue monitoring which is executed by the computer will produce significant data which the teacher of Math 110, Professor Desjarlais, and other educational psychology researchers are interested in analyzing to help interpret student learning within a group setting. The computer will classify the student conversations in two dimensions, both a domain-dependent dimension and a domain-independent dimension. The domain-dependent dimension is specific to student understanding of the particular activity being monitored and the process the group has used to come upon their solution to the problem. This dimension tracks the problem-specific realizations groups have displayed en route to their result. In the domain-independent dimension, classification of student dialogue is in regards to how the students are collaborating as members of a group. The fundamental notion in classifying the domain-independent dimension is to measure individual student participation in the group activity and the extent of contribution of ideas from a particular student.

The data that we have analyzed and classified this summer is from the Math 110 class, Quantitative Problem Solving, at Valparaiso University. This lower-level math course is to help students in their understanding of basic mathematical concepts and methods. The focal point of the course is problem solving, collaborative communication, and mathematical reasoning – all in a collaborative environment. Each day, students split into groups of three or four in order to solve a particular problem. Also, each student in the group is assigned one of four roles: reporter, presenter, moderator, and reflector. Each student, no matter which role he has, is to participate in the activity as well as complete the duties of his particular role. The reporter marks down the problem solving strategies and thought processes used by the group during the activity and is in charge of the group written report. The presenter is responsible for writing the group's results on the board and clearly explaining to the rest of the class the steps his group used in order to come to such results. The moderator is in charge of keeping the group on task during the activity, as well as mediating any tension that may arise between group members in the process. Finally, the reflector is to write a report reflecting on the activity, in regards to the actual work and the dynamics of the group, after the activity has been completed.

The particular Math 110 problem-solving activity we dealt with this summer is titled "Poison." Poison is a game where the students in each group are to form two teams. One team competes against the other team in the group. In order to begin, there are to be 20 tiles placed between the two teams. The rules of the game are as follows:

- Decide which team will play first.
- On your team's turn, your team is to remove either 1 or 2 tiles from the middle pile of tiles.
- The teams will alternate taking turns.

- The team that is forced to take the last tile - the *poison* tile - loses the game.

The group is to play the game a number of times, alternating which team plays first. The reporter in the group is to keep track of the moves and choices of the two teams while playing the game. Ultimately, the aim of Poison is to be able to determine how your team should play to force the other team to lose. This goal can be satisfied upon figuring out a pattern which occurs within the game.

The Math 110 Poison data we analyzed in our research consists of group reports, written by the reporter of a student group, and transcripts of cassette recordings of groups playing the game. The original group reports are written during the same class period that the groups play Poison. Though, the written reports we actually analyzed and classified this summer are a more polished and finalized version of each group's first attempt. The student reports are written in a narrative format, beginning with a description of the Poison problem. Then, a summary of the results and the overall thought processes the group used during the game, to come upon a strategy for forcing the other team to lose, is reported. There are 38 group reports which we examined throughout our research. The average length of each group written report is about 13 or 14 sentences.

The transcripts of the cassette recordings of groups in the process of playing Poison were the other data under analysis this summer. These cassette recordings are a taping of the students in the 2011 Valparaiso Experience in Research by Undergraduate Mathematicians (VERUM). Students in VERUM formed groups in order to play the Poison game during the first week of the summer program. Since there were three groups in the program, we had three different cassette recordings to analyze. Furthermore, the recordings are a log of the collaborative chat among students in the process of solving the Poison problem. Though, these cassette recordings differ from the written reports as the reports are more grammar and structurally-sound dialogues where as the cassette recordings are more informal and real-time dialogues of the problem-solving process. The average length of the transcripts of the cassette recordings is about 200 lines of dialogue each.

## II. Transcript Analysis

Transcript analysis was a very big component of our work this summer. To begin with, we had to type all 38 of the group reports into the computer since the reports are handwritten by students originally. It was necessary to key the reports into text files so the models we used during the next stage of our research process, data mining, could open and read these files in as input. Also at the beginning of the summer, we transcribed each of the cassette recordings of the three Valparaiso University REU groups playing Poison. There are three students in each of the groups and then on top of that there was an extra Poison participant in two of the groups. Professor Desjarlais was in an administrative position where occasionally she would join the dialogue of each of the groups, asking us where we were in the problem-solving process and prompting us to explain the realizations we had come to en route to finding the winning strategy.

A line of a transcript is marked with the label of A, B, C, D, E, or Z in order to keep track of if a different person is speaking or if the same person is responsible for consecutive lines of dialogue. Also, the specific label is important for indicating the particular role of who is talking or an action being performed. A, B, C, and D is the marking for either a student or extra participant speaking, E marks Professor Desjarlais is talking, and Z is the label for sounds of clapping, laughing, or playing (moving tiles around). The double slash symbol // is used in the transcripts to denote either two people are talking at once or one person interrupts another who was talking first. These transcripts can be found in the Appendix.

Once we played the Poison game as a group and wrote a report on the solution to Poison as well as examined other group written reports and transcribed the three cassette recordings, we had a pretty good idea of what understandings a group should come to during the Poison problem-solving process. We were then able to identify what we called "bits of realization," which are important concepts to understand the Poison problem. The 38 student written reports and 3 transcripts were analyzed a sentence at a time and tagged, human-classified, with a particular bit of realization if we deemed the sentence displayed such comprehension.

We also annotated the three transcripts in two other dimensions: conversational threads/tags and problem solving threads/tags. We got these threads and tags from the Virtual Math Team.[5]

**Bits of Realization** *Important concepts to understand the game of Poison*

| Code | Bits of Realization |
|---|---|
| 1 | 4 tiles is important |
| 2 | 2 and 3 are good tiles |
| 3 | You want to leave your opponent with 19 tiles |
| 4 | Going first gives you control of the game |
| 5 | You want to take 1 tile on your first move |
| 6 | 1, 4, 7, 10, 13, 16, 19 are the poison numbers |
| 7 | "Opposite" strategy |
| 8 | "3 pattern" |
| 9 | Wrong statements |
| 10 | Exploring |
| 11 | Playing the game |
| 13 | Making an observation |
| 14 | Clarifying observations |
| 15 | Clarifying rules |
| 16 | Exploring further versions of the game |
| 17 | Hypothesizing |
| 18 | There is a winning strategy |
| 19 | Filler |

A full description of the bits of realization can be found in the Appendix.

**Conversation Tags** *Categorize a line of transcript according to a dialogue move*

| Code | Conversation Tag |
|---|---|
| S | State |
| O | Offer |
| Rq | Request |
| Rg | Regulate |
| Rp | Respond |
| F | Follow |
| El | Elaborate |
| Et | Extend |
| Se | Setup |
| A | Agree |
| D | Disagree |
| C | Critique |
| E | Explain |
| Nc | No code |

A full description of the conversation tags can be found in the Appendix.

**Problem Solving Tags** *Categorize a line of transcript as a move related to the problem-solving strategy*

| Code | Problem Solving Tag |
|---|---|
| O | Orientation |
| S | Strategy |
| T | Tactic |
| P | Perform |
| Ch | Check |
| Re | Restate |
| Su | Summarize |
| Rf | Reflect |
| R | Result |

A full description of the problem solving tags can be found in the Appendix.

## III. Data Mining

Data mining was the other big component of our research project this summer. It is the means of finding and obtaining patterns from a given data set using mathematical methods. In our case, we were looking to be able to recognize bits of realization within a group written report or transcript using data mining. The data mining of each document (sentence) was done using a classifier, which for our purposes is defined as a program that reads a sentence and decides which bit of realization it is conveying. We wrote Python code in order to build our classifier for the classification of documents in the reports and transcripts.

Our classifiers were built out of 28 group reports - a total of 435 sentences. We then used the classifier based on a certain mathematical method to test 100 sentences, from the 10 other written reports, as well as the lines of dialogue from the transcripts. Eventually in the bigger overall project of our research project this summer, the goal is to build a classifier upon our fundamental classifier in order to categorize chat messages as students play in the online collaborative environment of Poison. In turn, these classifiers will be capable of keeping a log of what realizations a group of students have come to and also may be able to recognize if the students playing Poison have come to an impasse.

We wrote Python code to read in the documents (sentences) of the group written reports and produce a co-occurrence matrix. A co-occurrence matrix is a documents by words matrix which contains the frequency of words in particular documents as the entries of the matrix. The co-occurrence matrix is the actual data that the mathematical method of the classifier manipulates. The co-occurrence matrices we produced are based on 435 corpus documents, all from the group reports. A corpus is a collection or body of written texts and our corpus consists of 38 student written reports and 3 transcripts. Each of the 435 rows in the co-occurrence matrices are marked with a document label, consisting of the group report number and sentence number within the specific report, as well as the number of words in the particular sentence. The number of columns in a co-occurrence matrix corresponds to the number of distinct "words" in all of the 435 documents and is the reason why the following co-occurrence matrix is so sparse - containing lots of 0 entries.

| filename | sentence# | words/sentence | all | whoever | four | go | to | finally | tweek | advantage | you'd | every | continued | entire | did | turns | solution | leave | team | force | second | odd | what | deliberate | giving | ned | poison |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pwr1.txt | s1 | 21 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s2 | 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| pwr1.txt | s3 | 13 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s4 | 17 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s5 | 20 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s6 | 19 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s7 | 20 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| pwr1.txt | s8 | 32 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s9 | 13 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s10 | 15 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s11 | 14 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s12 | 27 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| pwr1.txt | s13 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr1.txt | s14 | 45 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s1 | 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| pwr2.txt | s2 | 13 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| pwr2.txt | s3 | 18 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s4 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s5 | 12 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s6 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s7 | 19 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s8 | 35 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| pwr2.txt | s9 | 29 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s10 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s12 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s13 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s14 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s15 | 23 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s16 | 20 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s17 | 23 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s18 | 18 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s19 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr2.txt | s20 | 30 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pwr3.txt | s1 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| pwr3.txt | s2 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Co-Occurrence Matrix

We analyzed the documents either as single words at a time, unigrams, or pairs of consecutive words together, bigrams. Here is an example of how to look at the document, "We found that 4 is a poison number," as a number of bigrams:

- we found
- found that
- that 4
- 4 is
- is a
- a poison
- poison number

Obviously, the number of columns in a co-occurrence matrix is much higher when either just bigrams or a combination of bigrams and unigrams are used than when just unigrams are used as the "words." Some words occur very frequently or infrequently within the reports and the written English language and in turn don't help in the classification of one document compared to another. Therefore in some of the co-occurrence matrices, we took out words which occurred in more than 60% of the total documents as well as words which occurred less than 4 times among all the documents. Finally, sometimes we employed a stop word list in order to remove the following 12 frequent words from the co-occurrence matrix:

· or · in · the · to · of · on
· a · an · at · and · it · is

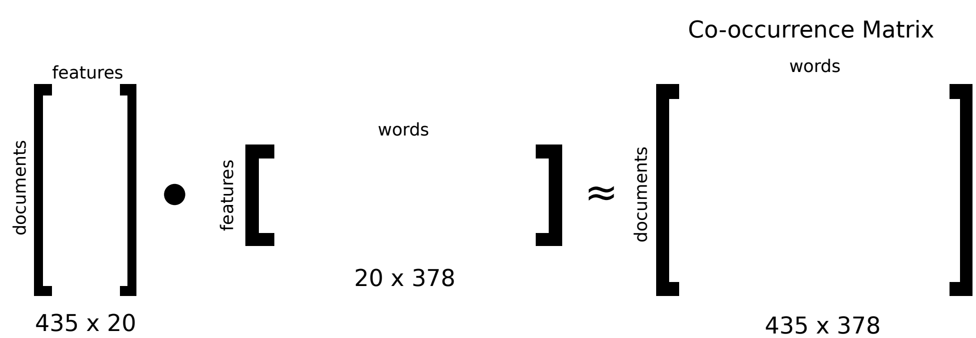
*Supervised vs. Unsupervised Methods of Classification*

When we do not give a program training data, it is called an unsupervised method. It categorizes the documents on its own, without us giving it a set of data that was pre-categorized for it to use as a foundation. Supervised methods of classification are when we do give the program data to train off of.

*Clustering*

Our first method of classification was unsupervised and used the K-Means Clustering option in WEKA.[1] We input our co-occurrence matrix into WEKA. The method takes each document and plots it in a multi-dimensional space based on Euclidean distance. It then assigns random centroids to the data and classifies the sentences to the centroid closest to it. We could control how many clusters we wanted and set a minimum number of documents in each cluster. We tried many different combinations of these variables. The method seemed to put a few documents into a cluster and throw the rest into an 'everything else' cluster. We found that it was not an effective method of classifying our data.

*Non-Negative Matrix Factorization*

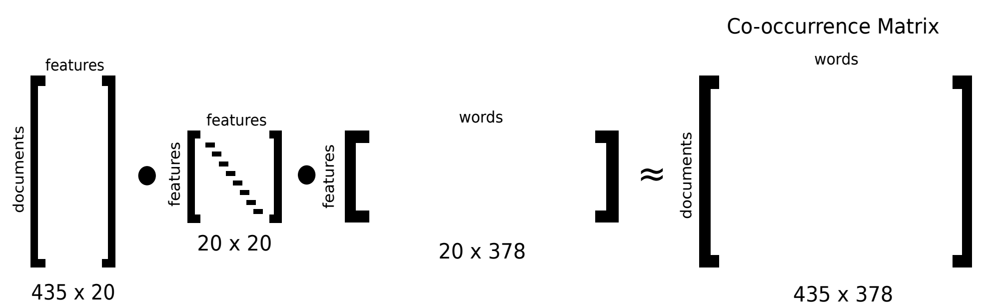



Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is another unsupervised method of classification. It decomposes the co-occurrence matrix into two matrices, creating a set of features along the way for which each document and each word is given a weight.[4] A feature is a set of words, ranked by prominence. If a sentence has words that are in the feature or words with similar context, it will receive a higher weight for that feature. Conversely, if a sentence has no words in the feature or words with similar context, it will receive a lower weight. We experimented with the number of features, comparing the sentences with the highest weights under each feature with the bits of realization we created. We found that 20 features matched most closely, producing the most features whose sentences related to bits of realization that we hand-classified.

One notable aspect of the NMF method is that all the entries in the matrices it produces are positive numbers. This is more applicable to our research (it would be strange to reason that a word has a negative context in a given situation). Another attribute of this method is that it does not produce a unique result. We can feed it the same co-occurrence matrix as many times as we want and it will produce slightly different features and weights every time. This makes it more difficult to test with new, unknown documents. Not only that, but it takes a long time for the NMF method to test new sentences. In order to create a vector for a new sentence, the program must solve an overdetermined set of linear equations that involve 20 variables and hundreds of equations. Multiply that by 100 new sentences. Our program took hours to produce one set of results (unigram, bigram, both).

*Singular Value Decomposition*




Singular Value Decomposition

Singular Value Decomposition (SVD), another unsupervised method of classification, is very similar to the NMF method conceptually. It still factors the co-occurrence matrix into matrices and creates features for us to compare with our bits of realization. SVD factors the matrix into three matrices, unlike NMF's two. The middle matrix is a square, diagonal matrix of scalar weights and is called the Singular Value Matrix. The two matrices that sandwich it are each orthogonal (multiply it by its transpose and it produces the identity matrix).

The SVD method produces a unique result and is faster to run than the NMF method. This is because to compute the vector of a new sentence, the method need only take the columns of one of the three matrices, corresponding to each word in the sentence, sum them, and divide them by the singular value matrix.

One aspect of the SVD method that some computer scientists view as a downfall is that it produces negative entries in the matrices. Those computer scientists claim that this makes the SVD method less applicable to this type of research, but we don't consider this a flaw or rule it out because of this.

*Latent Semantic Analysis*

Latent Semantic Analysis (LSA) is a supervised method of classification.[3] First we created bundles, which are lists of sentences each having to do with a specific bit of realization. Then we took an unknown sentence and calculated the cosine between it and every sentence in every bundle (using both the NMF and SVD models to get the vectors). A cosine close to 1 means that the two sentences have words and/or content in common. A sentence close to 0 means that the two sentences convey different ideas and have few or no words in common. Therefore we took the highest cosine and whichever bundle it was in, we labeled the new sentence as containing that correlating bit of realization. Even though we use the NMF and SVD models in this method, it is labeled as supervised because we manually chose which sentences went in each bundle, and therefore gave the method outside information to help it come to a result.

## IV. Results

*Unsupervised Testing*

We first tested student written reports using an unsupervised method. We made 6 different classifiers built off of 28 student written reports (435 sentences):

- Non-Negative Matrix Factorization with Unigrams, removing words that occurred in less than 4 documents, and over 60% of documents
- Non-Negative Matrix Factorization with Unigrams, removing words that occurred in less than 4 documents, and over 60% of documents, and removing stop words
- Non-Negative Matrix Factorization with Bigrams
- Singular Value Decomposition with Unigrams, removing words that occurred in less than 4 documents, and over 60% of documents
- Singular Value Decomposition with Unigrams, removing words that occurred in less than 4 documents, and over 60% of documents, and removing stop words
- Singular Value Decomposition with Bigrams

We used each model to test 10 student written reports (113 test sentences) that were manually tagged with bits of realization. Testing was done by analyzing the 'features' created by each model, and evaluating if they conveyed any bits of realization. The bits of realization that were found within features can be found in Table 1 below. We ran each test sentence through our models as if they were sentences already apart of the model, and found which features would be most highly associated with each test sentence. The top 3 features were evaluated for each test sentence. We then looked at which sentences we would've expected to show up within each feature versus how many actually did show up in each feature.

| | **Unigrams** w/o freq./infreq. words | **Unigrams** w/o freq./infreq. words w/o stop words | **Bigrams** |
|---|---|---|---|
| **NMF** | 87% top feature 15 | 86% top feature 14 | |
| | "Opposite Strategy" (13/13,63) | — | "Opposite Strategy" (11/13,19) |
| | — | "Clarifying Rules" (16/27,8) | "Clarifying Rules" (23/27,23) |
| | "Poison #s" (12/27,2) | — | "Poison #s" (12/27,10) |
| | — | "4 tiles is important" (5/9,15) | "4 tiles is important" (5/9,15) |
| **SVD** | "Poison #s" (8/27,19) | — | "Poison #s" (18/27,43) |
| | "Opposite Strategy" (11/13,43) | "1 tile first move" (0/4,1) | — |

Table 1: Unsupervised Results - 10 student written reports

Table 1 can be read like this:
"Bit of Realization" ( Computer correctly classified sentences containing 'Bit of Realization'/Human tagged sentences containing 'Bit of Realization', False Positives)

So "Opposite Strategy"(13/13,63) can be read as:
Opposite strategy was found within the NMF model, and the model correctly classified 13 out of the 13 Opposite Strategy sentences, but the computer falsely classified 63 sentences as having opposite strategy.

As shown in Table 1, results were not good. There were never more than 4 bits of realization represented well within a model. Often, models could correctly classify most or all sentences but had a very large number of false positives. Conversely, many models had very few false positives, but also missed most or all of the correct sentences. There was also a similar problem with the NMF method as in the Clustering. In both of the unigram models, a very large percentage of sentences (87% and 86%) had one feature as one of the top features. Grouping a large amount of sentences like this was solved in the SVD models, but the models didn't group as many bits of realization into features.

*Supervised Testing*

We then tested 10 student written reports using a supervised method.

We used 6 classifiers built off of 28 student written reports (435 sentences), slightly varying from the previous 6 models:

- Non-Negative Matrix Factorization with Unigrams, removing words that occurred in less than 4 documents, and removing stop words
- Non-Negative Matrix Factorization with Bigrams
- Non-Negative Matrix Factorization with both Unigrams and Bigrams
- Singular Value Decomposition with Unigrams, removing words that occurred in less than 4 documents, and removing stop words
- Singular Value Decomposition with Bigrams
- Singular Value Decomposition with both Unigrams and Bigrams

We again used each model to test 10 student written reports (113 sentences) that were manually tagged with bits of realization. We used Latent Semantic Analysis as described previously, to compare each test sentence to bundles of sentences, one bundle for each bit of realization. Comparison was made by computing the cosine between each test sentence and each sentence in each bundle (sentences are represented by vectors). We did not have a bundle for bit 19 (Filler), so those sentences were excluded from our analysis, bringing the total number of test sentences down to 100 sentences. The highest cosine value associated with each test sentence was chosen, and whichever bundle this sentence came from was deemed to be the computer tagged bit of realization. As mentioned before, using bundles turns this into a supervised method of learning.

Generally, the results using this method were much better than the unsupervised method, and can be found in Table 2. Our classifier that did the best was the SVD model, which achieved a 55% accuracy when looking at all bits of Realization. Bits 6, 11, and 15 were the bits of realization with the most manually tagged sentences (19, 13, and 27 respectively), and when looking at only these sentences, our models did even better, the SVD bigram model achieving 68% accuracy. Individual results among these dimensions are also displayed, and generally are even better. The variety of values across bit 11 is unexpected, but at this time we have no explanation for this. We performed the ChiTest on these three bits for each model. A ChiTest produces

| | Computer Correct | | | Top 3 Bits | | |
|---|---|---|---|---|---|---|
| | All Bits Results | Top 3 Bits Results | ChiTest Results | Bit 6 (19) | Bit 11 (13) | Bit 15 (27) |
| NMF Unigrams | 47% | 61% | .002972 | 58% | 31% | 78% |
| NMF Bigrams | 45% | 58% | .026798 | 37% | 38% | 81% |
| NMF Both | 48% | 64% | .023791 | 52% | 54% | 78% |
| SVD Unigrams | 51% | 66% | .000292 | 52% | 86% | 85% |
| SVD Bigrams | 55% | 68% | .027772 | 63% | 15% | 96% |
| SVD Both | 53% | 59% | .003231 | 42% | 0% | 100% |

Table 2: Supervised Results - 10 student written reports

a p-value, and $p < .05$ is typically considered to be statistically significant. For all 6 models we computed a p-value less than .05, and so we can assume our results are not due to random chance.

After finding these results, we wanted to use our models to test our transcript dialogues. Testing these transcipts could hopefully give similar results to testing chat messages, which is the long-term goal of this project. We decided to use our supervised method, but right away foresaw some problems. A 'real-time' dialogue is very different from a student written report. While the written reports are a summary of what students did, the transcripts are a conversation of students actually solving the problem. Much of the vocabulary from the transcripts is not present in the written reports, thus trying to build vectors to represent transcript sentences may not work well. Another issue is that many of the utterances from the transcripts consist of only one word ("Yeah", "Right", "OK"), and these utterances don't have any bigrams. Finally, in spoken dialogue, especially in a problem solving context, people often finish each others' thoughts or ideas. So finding bits of realization in a single utterance may prove to be difficult.

Even with these problems in mind, we still did the same supervised testing, and the results can be found in Table 3. The three transcripts consist of 562 sentences.We used the same 6 models that were used in testing the written reports. The results are lower, which was expected, and the SVD model with both unigrams and bigrams had the best results.

| | Overall Results |
|---|---|
| NMF Unigrams | 15% |
| NMF Bigrams | 9% |
| NMF Both | 19% |
| SVD Unigrams | 15% |
| SVD Bigrams | 7% |
| SVD Both | 21% |

Table 3: Supervised Results - 3 Transcripts

| | Overall Results |
|---|---|
| NMF Unigrams | 15% |
| NMF Bigrams | 12% |
| NMF Both | 12% |
| SVD Unigrams | 15% |
| SVD Bigrams | 14% |
| SVD Both | 17% |

Table 4: Supervised Results - 3 Transcripts using Sliding Windows

We have 2 options in mind in how to overcome the problems with testing transcripts. The first idea is to use Thread-Based Analysis. This involves the conversational tags and the problem solving tags we manually placed on the transcripts. These tags come from Virtual Math Team[5], a group of students who solve math problems in an online chat environment. Conversational Analysis is used to track the give and take, and the adjacency pairs of conversation. The other solution is to use a sliding window technique. Instead of taking each document to be a sentence or utterance from the transcripts, we could take some number of utterances. This would fix both the problem of having only one word utterances, and also possibly represent students finishing each others' ideas. We did initial testing with this method taking how ever many number of utterances created at least 10 words to be a document. These results (shown in Table 4) did not prove to be any better than our initial supervised testing, but we are unsure why.

## V. References


1. Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009); *The WEKA Data Mining Software: An Update*; SIGKDD Explorations, Volume 11, Issue 1.

2. Landauer, Thomas K., and Susan T. Dumais. "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and.." *Psychological Review* 104.2 (1997): 211. Print.

3. Landauer, Thomas, Danielle McNamara, Simon Dennis, and Walter Kintsch, eds. *Handbook of Latent Semantic Analysis.* Mahwah, NJ: Lawrence Erlbaum Associates, 2007. Print.

4. Segaran, Toby. *Programming Collective Intelligence: Building Smart Web 2.0 Applications.* Beijing: O'Reilly, 2007. Print.

5. Stahl, Gerry. *Studying Virtual Math Teams.* New York, NY: Springer Press, 2009. Print.

## VI. Appendix

**Bits of Realization** *Important concepts to understand the game of Poison*

**15.** *Clarifying rules*
This is when the students repeat the given rules to the game. Usually in the form of a question.

Examples:

"In the game of Poison, you start with 20 tiles."
"It says play this game a number of times, alternating who plays first."
"The object is not to take the last tile?"

**11.** *Playing the game*
Transcripts only. Student commentary while the game is played.

Examples:

"So who wants to go first?"
"[Sounds of playing]"
"Let's take 1."

**10.** *Exploring*
Instead of playing a full game, the students will suggest starting with a fewer number of tiles. This is also labeled when students add tiles to the pile.

Examples:

"Do we want to break it down into a smaller game of less than 20 tiles?"
"If you have 9, can you win from there?"
"Ok um, so if we add these 2 back in here, you can take those 2."

**2.** *2 and 3 are good tiles*
Once the game is reduced to two or three tiles, the outcome is determined. Whichever team picks from two or three tiles will win. Players want to be left with two or three tiles on their turn.

Examples:

"So, if you have 3, if you're picking, that's good for you."
"When there's 3 left, and it's not your turn, you should lose, because the other person should take 2."
"Good: 2, 3."

**1.** *4 tiles is important*
Once the game is reduced to four tiles, the outcome is determined. Whichever team has to pick from four tiles will lose (When you pick from four tiles, you leave the other team with three or two. See *2 and 3 are good tiles* above). Players do not want to be left with four tiles on their turn.

Examples:

> "Ok so you want to get it to 4, that's kind of a stable point where you can force them to lose."
> "So for 4, so if it's 4, and you're picking, then you can't win."
> "Ok, so when there was 4 left, the person who was next to go, loses."

**6.** *1, 4, 7, 10, 13, 16, 19 are the poison numbers*
These are the numbers you want to avoid while playing the game. In a winning strategy, a team would force its opponent to pick from these numbers.

Examples:

> "Ok so basically the way you want to play is you want to kind of get the other team to this point of 4, 7, 10, whatever."
> "So you take the number of tiles you need each turn to avoid ending up with 4, 7, 10, 13, 16, or 19 tiles."
> "When you leave your opponent with these bad numbers you are in control of the game."

**7.** *"Opposite" strategy*
This is the strategy used by players to keep their opponent picking from the poison numbers. It decreases the number of tiles in the pile by three every round of play. If the team's opponent picks one tile, the team will pick two, and vice versa.

Examples:

> "If you take 1 or 2, then you can take 1 or 2 to counterbalance that."
> "Because if they pick 1, the next team can pick 2 and leave you with the last 1, and if you pick 2, then the last team can pick 1 and leave you with the last 1."
> "Basically you want to take the opposite of what they're doing."

**8.** *"3 pattern"*
This is assigned when the students notice that the poison numbers increase by three, or when they point out that in a winning strategy, they want to remove three tiles every round of play.

Examples:

"Group of 3 take the remainder."
"So it seems to be going up by 3s."
"If the total is ever 1 mod 3 at the start of your turn, then you lose."

**18.** *There is a winning strategy*
This is when the players allude to or explicitly state the fact that there is a strategy that guarantees a team's victory.

Examples:

"Well you can force a win."
"You want to be able to say, regardless of what the other team does, you can win."
"We just want to know how to play to make the other person lose."

**4.** *Going first gives you control of the game*
This is assigned when the players allude to or explicitly state that the winning strategy involves going first or that the team that goes first can manipulate the game.

Examples:

"So whoever goes first, wins, if they play it smart."
"Whoever goes first always has control."
"By going first, you can get the other players down to the bad numbers."

**3.** *You want to leave your opponent with 19 tiles*
The players want to force their opponents to pick from the poison numbers, the first of which is 19.

Examples:

"When we realized that number 19 was a poison number we knew that whichever team went first won because at the beginning they could take 1 tile leaving the opposing team with 19 tiles which is a losing number."
"Force the other team to have 19."
"To start the first team should take 1 tile to make it 19."

**5.** *You want to take 1 tile on your first move*
As part of the winning strategy, the players should point out not only that *going first gives you control of the game* but also that the team's first move

should be to remove one tile. This move precedes the *opposite strategy* when a team is playing to win the game.

Examples:

"If you go first, you want to take 1."

"In the end, the guarantee that you can win is taking the first move and taking 1 tile."

"If you grab the first tile you can basically control the game."

**17.** *Hypothesizing*

Transcripts only. This is when the students are considering aloud all the options beginning with a certain number of tiles. It is in an attempt at finding a pattern or strategy to win the game from that starting point.

Examples:

"So then if you can take away 2 tiles, um, you want to get your opponent to 7 tiles."

"If you guys took 2, we'd take 1, you guys would take 1 and you guys would lose."

"But if you don't go first, how can you win?"

**13.** *Making an observation*

Transcripts only. This usually follows hypothesizing, when the players agree on the appropriate move to make when faced with a specified number of tiles.

Examples:

"Well the 1 thing I noticed is that if there's 6 tiles left and it's your turn, you can force the other team to lose."

"When there's 8, you'd want to take 1."

"If you pick 1 or 2 then the other person has the 9 or 8 to make it 7, so 10 you lose."

**15.** *Clarifying observations*

Transcripts only. This is assigned when students review the realizations they have already come to. Usually in the form of a question.

Examples:

"You are adding 1 tile at a time and seeing how to force a win, right?"

"Wait, can we go over 6 again?"

"What did you do last time?"

**9.** *Wrong statements*

This is assigned when students make statements that seem like a strategy or a bit of realization listed above, but are faulty.

Examples:

"You want to make sure your team always has an odd number."
"1, 4, 7, 12, 15, 18 are the poison numbers."
"You want to take 2 on the first move."

**16.** *Exploring further versions of the game*
After a group of students has figured out the wining strategy to Poison and there is still class time left, the professor might suggest that they play a revised game, where the starting number of tiles is different or the turn-taking options are expanded.

Examples:

"What about if we started with 30 tiles?"
"What if you could choose 1, 2, or 3 tiles on your turn?"
"It'd change a lot if you could choose like 0 or 3 tiles instead of just 1 or 2."

**19.** *Filler*
Transcripts only. This is when the students say something not related to the problem-solving strategy.

Examples:

"Yeah," "Ok," "Right," "Um"
"Do we have a lot of time?"
"[Sounds of laughing]"

**Conversation Tags** *Categorize a line of transcript according to a dialogue move*

**O** *Offer*

A line which introduces content which has not been previously discussed.

Example:

"So now let's play, and see if we can, now that we have this guide, see if we can force a win."

**Rq** *Request*

A line in which any type of request is made to another group member.

Example:

"So there were 6, right?"

**Rg** *Regulate*

A line about a turn-taking or another activity that is being or is going to be performed.

Example:

"Ok, so who wants to, should we go first?"

**S** *State*

A statement that is not as specific as the ones listed above.

Example:

"We have an even number."

**F** *Follow*

A response to another group member that does not explicitly agree or disagree.

Example:

"Ok," "Right," "Yeah"

**El** *Elaborate*

A line that follows a line by the same speaker and adds to it.

Example:

"[If I take 2, that would leave you with 3, and I lose.] So I'd want to take 1."

**Et/Et2** *Extend*

The completion of a fragmented statement spoken by (Et) the same person or (Et2) a different person.

Example:

"[So it's our turn and we'll take one.] And then we lose."

**Se/Se2** *Setup*

The beginning of a fragmented statement spoken by (Se) the same person or (Se2) a different person.

Example:

"Add a ninth one, you want to take [2 to give them 7.]"

**A** *Agree*

A line in which the speaker explicitly agrees with a previous statement.

Example:

"[So then what about 13, 13 you lose?] 13 you lose."

**D** *Disagree*

A line in which the speaker explicitly disagrees with a previous statement.

Example:

"[It's paired in 4s, right?] It looks like 3s."

**C** *Critique*

A line which not provides an alternative to a previous statement.

Example:

"[It looks like 3s.] Well it's 4s and then its 3s after that I think."

**E** *Explain*

An explanation is provided, usually in response to a request.

Example:

"[What should you do at 8?] I want to take 1, to leave them with 7."

**Rp** *Respond*

A response to another group member that is not as specific as the ones listed above.

Example:

"I have no idea"

**Nc** *No code*

A line that cannot be assigned to any of the above tags.

Example:

"Ok, um, ok."

**Problem Solving Tags** *Categorize a line of transcript as a move related to the problem-solving strategy*

**O** *Orientation*

A line in which the rules of the problem are repeated.

Example:

"It says play the game a number of times, alternating who plays first."

**S** *Strategy*

A line in which a long-term problem-solving method is suggested.

Example:

"Ok let's assume that everyone always plays optimally."

**T** *Tactic*

A line in which a short-term problem-solving method is suggested.

Example:

"So if there's 10, what can we do from there?"

**P** *Perform*

A line in which a chosen strategy or tactic is executed.

Example:

"If you pick 1 or 2 then the other person has the 9 or 8 to make it 7."

**Ch** *Check*

A line in which a strategy or performance is checked or evaluated.

Example:

"Wait can we go over 6 again?"

**Re** *Restate*

A line in which a result or performance is repeated.

Example:

"They were playing first and they took 2?"

**Su** *Summarize*

A line in which a problem-solving set is repeated in a condensed fashion.

Example:

"Because if they take 1, they'd leave you with 8 to choose from, we've already laid out the strategy to win if you have 8, you can just take 1."

**Rf** *Reflect*

A line in which a problem-solving set is reintroduced with reasoning.

Example:

"Because if your opponent is ever stuck in a place where they have 4, 7, whatever, then you can make them lose."

**R** *Result*

A line in which the outcome of a performance is stated.

Example:

"Ok, so when there was 4 left, the person who was next to go, loses."

Transcripts tagged 3 ways:
1. Bit of Realization
2. Conversational Tag and corresponding thread (C and CT)
3. Problem-Solving Tag and corresponding thread (P and PT)

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr001-001 | A | 11 | | Rg | | | So we go first. |
| tr001-002 | A | 11 | | Rg | | | So how many tiles would you like to take? |
| tr001-003 | B | 11 | 002 | Rg | | | OK Let's take... |
| tr001-004 | B | 11 | | Rg | | | Do you want to write the teams down first? |
| tr001-005 | C | 11 | | Rg | | | You guys are team 1, so you get to go first. |
| tr001-006 | C | 11 | | Rg | | | Should I just write how many you take? |
| tr001-007 | B | 19 | 006 | F | 006 | | Yeah |
| tr001-008 | A | 19 | 006 | F | 006 | | Yeah |
| tr001-009 | C | 19 | 006 | F | 006 | | Alright |
| tr001-010 | B | 11 | | Rg | | | Let's take 1 |
| tr001-011 | C | 11 | 010 | Rg | | | Go ahead |
| tr001-012 | Z | 19 | | Rg | | | Sounds of mumbling |
| tr001-013 | Z | 19 | | Rg | | | Sounds of laughing |
| tr001-014 | Z | 19 | | Rg | | | Sounds of playing |
| tr001-015 | D | 15 | | Rq | | O | The object is not to take the last tile? |
| tr001-016 | B | 19 | 015 | F | 015 | | Right |
| tr001-017 | C | 19 | 015 | F | 015 | | Right |
| tr001-018 | A | 11 | | Rg | | | Oh no, you have to take it. |
| tr001-019 | A | 11 | | Rq | | | And we lose, and make sure that sums to... |
| tr001-020 | C | 11 | 019 | Rp | | | Yeah I'm not sure it does... |
| tr001-021 | C | 11 | 020 | El | 020 | | So, 2, 4, 6, 8, 2, 4, 6, 8, 10, 2, 4, 6, 8, 10 nope that's |
| tr001-022 | A | 11 | 021 | Rg | | | Alright so our data is correct |
| tr001-023 | B | 15 | | O | | O | It says play this game a number of times, alternating who plays first. |
| tr001-024 | B | 19 | 023 | Rq | | | Do we need to do that? |
| tr001-025 | B | 10 | 024 | O | 024 | T | Or do we want to break down it into a smaller game of less than 20 tiles? |
| tr001-026 | D | 13 | | O | 025 | R | Well the one thing I noticed is that if there's 6 tiles left and it's your turn, you can force the other team to lose. |
| tr001-027 | B | 19 | 026 | F | 026 | | Alright |
| tr001-028 | A | 19 | 026 | Rq | | | How? |
| tr001-029 | D | 7 | 028 | E | 028 | R | If you take 2, then whatever you do on the next turn, you can do the opposite to leave 1. |
| tr001-030 | B | 7 | 028 | E | 028 | R | If you take 1 or 2, then you can take 1 or 2 to counter balance that |
| tr001-031 | A | 19 | 030 | F | 030 | | OK |
| tr001-032 | C | 19 | 030 | F | 030 | | OK |
| tr001-033 | C | 17 | 029 | Rq | 029 | T | So if I take 2, whatever they do... |
| tr001-034 | B | 1 | | O | 033 | R | So basically if the other team ends up 4 left, then you can win. |
| tr001-035 | D | 19 | 034 | A | 034 | | Yes |
| tr001-036 | B | 1 | 034 | El | 034 | Re | And that's if the other team ends up with 4 left |
| tr001-037 | B | 19 | 036 | F | 036 | | OK |
| tr001-038 | A | 19 | | Rg | | | We could maybe abbreviate opponent as OPP or something. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr001-039 | A | 19 | | Rg | | | Whatever, you might be writing a lot. |
| tr001-040 | B | 19 | | Nc | | | So yeah, um |
| tr001-041 | Z | 19 | | Rg | | | Sounds of mumbling |
| tr001-042 | C | 19 | | Nc | | | OK, um |
| tr001-043 | B | 19 | | Nc | | | Oh boy |
| tr001-044 | A | 19 | | Nc | | | We don't need grammar. |
| tr001-045 | B | 1 | | Rq | 034 | Re | Um so, if they 4 left you can win have how can you get it so that.. |
| tr001-046 | D | 13 | | O | | R | If you have 5 or 6 on your turn, you can either take 1 or 2 to get it to that situation. |
| tr001-047 | B | 1 | 034 | El | 034 | Re | Ok so you want to get it to 4, that's kind of a stable point where you can force them to |
| tr001-048 | C | 7 | | S | 047 | R | You took 1, then we'd take 2, if you took 2 then we'd take 1 |
| tr001-049 | B | 17 | | Se | | T | OK um, so if we add these 2 back in here, you can take those 2. |
| tr001-050 | B | 17 | 049 | Se | 049 | T | If you add 1 back in there, you can take those. |
| tr001-051 | B | 13 | 049 | O | 049 | R | If you end up with 7, the other team can force you to lose by taking, wait, |
| tr001-052 | C | 19 | | Se | | | Oh boy. |
| tr001-053 | C | 19 | 052 | Rg | | | I don't know what to write down |
| tr001-054 | B | 19 | | Nc | | | Ok um, ok |
| tr001-055 | A | 10,14,18 | 051 | Rq | 051 | Rf | I can see what your pattern is, you are adding 1 tile at a time and seeing how to force a win, right? |
| tr001-056 | B | 19 | 055 | F | 055 | | Right. |
| tr001-057 | A | 19 | | Se | | | So, maybe let's organize this a little bit. |
| tr001-058 | A | 19 | 057 | Rg | | | The process in your data collection |
| tr001-059 | B | 19 | 058 | Rq | | | Yeah, OK so do we want to say... |
| tr001-060 | C | 1 | | Rq | 034 | Ch | So with 4 tiles, we would take we when it's our turn, 4 tiles our turn, we would want to take. |
| tr001-061 | D | 1 | 060 | E | 060 | Re | When there's 4 tiles on our turn, we lose |
| tr001-062 | A | 19 | | Se | | | So let's have number of tiles on your turn? |
| tr001-063 | A | 19 | 062 | Se | 062 | | Or on your opponents turn? |
| tr001-064 | A | 19 | 063 | Rg | | | I think it's easiest on your turn |
| tr001-065 | B | 19 | 063 | Rg | | | Lets say your turn because we want to win right? |
| tr001-066 | A | 19 | | Nc | | | And then you can make the choice |
| tr001-067 | C | 19 | 065 | F | 065 | | Alright |
| tr001-068 | A | 19 | 066 | Rq | | | Right? |
| tr001-069 | C | 19 | | Nc | | | I think my eraser is done |
| tr001-070 | B | 19 | 069 | Rg | | | You could just flip to the next sheet, since we'll be data collecting |
| tr001-071 | A | 19 | | Rg | | | Do we have a lot of time, or no? |
| tr001-072 | E | 19 | | Rq | | | Where are you at? |
| tr001-073 | B | 10 | | E | | Re | We've broken down to a smaller game |
| tr001-074 | E | 19 | | F | | | OK |
| tr001-075 | B | 19 | | El | | | A few times |
| tr001-076 | E | 19 | | Rq | | | Good, how many tiles? |
| tr001-077 | B | 19 | | E | | | Um, 4 right now |
| tr001-078 | E | 19 | | Rq | | | What'd you figure out? |
| tr001-079 | A | 12 | | E | | | 1 |
| tr001-080 | Z | 19 | | Rg | | | Sounds of laughing |
| tr001-081 | A | 12 | | El | | | We're going to work up to 2 in a few minutes here. |
| tr001-082 | C | 1 | | Rg | | | Ok, 4 tiles for our opponent |
| tr001-083 | B | 19 | 082 | Rg | | | Well no, let's do it on your turn |
| tr001-084 | C | 19 | | Nc | | | So then |
| tr001-085 | B | 19 | 083 | El | 083 | | So then these will be just on your turn, how many tiles are left |
| tr001-086 | A | 14 | | Rq | 034 | Ch | So if there are 4 tiles, can you win or lose? |
| tr001-087 | D | 18 | 086 | Se | 086 | | Well assuming that everyone plays optimally from then on |
| tr001-088 | A | 18 | 087 | O | | S | Ok let's assume that, lets assume that everyone always plays optimally |
| tr001-089 | B | 19 | 088 | F | 088 | | OK |
| tr001-090 | D | 1 | 087 | E | 034 | Re | And if it's 4 we lose. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr001-091 | D | 13 | 090 | El | 090 | R | 5 or 6 you win. |
| tr001-092 | Z | 19 | | Nc | | | Sounds of clapping |
| tr001-093 | D | 13 | 090 | El | 090 | R | 7 you lose because whatever choice you make, you leave your opponent with 5 or 6 |
| tr001-094 | C | 1,13 | 093 | F | 093 | Su | Ok so 4 you lose, 5 and 6 wins, 7 lose |
| tr001-095 | B | 19 | 094 | F | 094 | | mhm |
| tr001-096 | D | 19 | 094 | F | 094 | | mhm |
| tr001-097 | D | 13 | | O | | R | 8... you can leave your opponent with 7, so 8 you win. |
| tr001-098 | B | 6 | 097 | Et2 | 097 | Rf | Because if your opponent is ever stuck in a place where they have 4, 7, whatever, then you can make them lose. |
| tr001-099 | B | 17 | 098 | El | 098 | Rf | So then if you can take away 2 tiles, um, you want to get your opponent to 7 tiles. |
| tr001-100 | D | 19 | | O | | | So, 9 |
| tr001-101 | B | 13 | 099 | El | 099 | R | So either 8 or 9 you should be able to win by taking away 2 tiles or 1 tile to get to 7 |
| tr001-102 | D | 19 | 101 | F | 101 | | Yep |
| tr001-103 | C | 19 | 101 | F | 101 | | Alright |
| tr001-104 | B | 13 | | O | | Rf | And then I think that's a pattern that's going to continue throughout the game |
| tr001-105 | D | 19 | 104 | F | 104 | | Makes sense |
| tr001-106 | C | 13 | | O | 104 | R | So 10 we lose |
| tr001-107 | B | 19 | 106 | F | 106 | | Right |
| tr001-108 | A | 10 | | O | 104 | T | So maybe we should build this up to 20 |
| tr001-109 | C | 19 | 108 | F | 108 | | OK |
| tr001-110 | D | 13 | | O | 106 | R | Well then at 11 or 12 you win. |
| tr001-111 | D | 13 | 110 | El | 110 | R | 13 you lose. |
| tr001-112 | D | 13 | 110 | El | 111 | R | 14, 15, wins. |
| tr001-113 | B | 14 | | S | | Su | So kind of what we have here is we have 1 tile, if you get this 1 you lose. |
| tr001-114 | B | 14 | 113 | El | 113 | Su | If you have these 2 you can make the opponent lose |
| tr001-115 | C | 13 | | Rq | 112 | R | So... 20 we win? |
| tr001-116 | A | 18 | 115 | E | 115 | | Well you can force a win. |
| tr001-117 | A | 19 | | O | | S | So, if you start, so, look at that. |
| tr001-118 | A | 15 | 117 | Rq | | O | 20 is the starting point, right? |
| tr001-119 | A | 4 | 117 | El | 117 | P | So if you go first.. |
| tr001-120 | D | 5,18 | 119 | A | 119 | P | Yeah, you take 1 and then you continue to play optimally. |
| tr001-121 | B | 6 | | O | | Su | OK so basically the way you want to play is you want to kind of get the other team to this point of 4, 7, 10, whatever |
| tr001-122 | C | 19 | 121 | F | 121 | | OK |
| tr001-123 | B | 19 | | Se | | | So |
| tr001-124 | Z | 19 | | Rg | | | Sounds of playing |
| tr001-125 | B | 9 | 123 | Rq | | | So it's almost paired in... no... it's paired in 4s right? |
| tr001-126 | A | 8 | 125 | D | 125 | | It looks like 3s. |
| tr001-127 | B | 14,8 | 126 | C | 126 | | Well its 4s and then its 3s after that I think |
| tr001-128 | C | 19 | 127 | Rq | | | What? |
| tr001-129 | B | 6 | 127 | El | 127 | Su | OK so this is 4, 7, 10, 13 |
| tr001-130 | A | 11,18 | | O | | S | So now lets play, and see if we can, now that we have this guide, see if we can, see if we can force a win. |
| tr001-131 | C | 11 | 130 | Rq | | | OK, who goes first? |
| tr001-132 | A | 11 | 131 | Rg | | | You go first this time |
| tr001-133 | B | 18 | 132 | Et2 | 132 | T | And try and force a win |
| tr001-134 | A | 11 | | Rg | 133 | P | So you take 1, and now, lets take 2 |
| tr001-135 | C | 11 | | Rg | 133 | P | You take 1 |
| tr001-136 | A | 11 | | Rq | | | Let's see, so how many are left? |
| tr001-137 | C | 11 | 136 | E | 136 | | 17 |
| tr001-138 | D | 11 | 137 | Rg | | | So we want to... |
| tr001-139 | C | 11 | 137 | Rg | 137 | P | We want 16 so we take 1. |
| tr001-140 | C | 19 | | Rg | | | I should be writing this down, T 1, T 2. |
| tr001-141 | A | 19 | 139 | F | 139 | | Ok |
| tr001-142 | A | 11 | | Rg | | Ch | So you took |
| tr001-143 | C | 11 | 142 | Rg | 142 | Ch | We took 1 |
| tr001-144 | A | 11 | 140 | Rg | | | No no no switch it, so it goes by who goes first |
| tr001-145 | C | 11 | 143 | Rg | | | Ch So 1, |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr001-146 | A | 11 | 145 | Rg | | | Ch And then we took 2 |
| tr001-147 | C | 11 | 146 | Rg | | | Ch And then we took 1 |
| tr001-148 | A | 19 | | Nc | | | Ok so |
| tr001-149 | B | 11 | | Rg | | P | So say we took 1 so then you guys can... |
| tr001-150 | C | 11 | 149 | Rg | | | 1 |
| tr001-151 | B | 11 | | Rg | | P | You take 2 |
| tr001-152 | D | 11 | 151 | Rg | | P | You take 2 |
| tr001-153 | B | 11 | 152 | F | 152 | | 2 |
| tr001-154 | D | 8 | | O | | R | Just always empty out that block of 3 |
| tr001-155 | A | 19 | 154 | F | 154 | Ch | Right |
| tr001-156 | B | 19 | 154 | F | 154 | | Ok |
| tr001-157 | D | 13 | 154 | El | 154 | Rf | Because then when you do it and you're last, you only have 1 left over. |
| tr001-158 | B | 11 | | O | | | Ok so it's our turn, and we'll take 1 |
| tr001-159 | A | 11 | 158 | Et2 | 158 | | 2, 1, 2, 1, and we're stuck |
| tr001-160 | B | 11 | 159 | Et2 | 158 | | And then we lose |
| tr001-161 | A | 11,8 | 154 | O | | Su | Ok so yeah I think it's a really great way of visualizing it, when you put these groups of 3 here, and then you take the final tile, you always no, you know, you're going to have to take the remainder of the group. |
| tr001-162 | Z | 19 | | Rg | | | Sounds of playing |
| tr001-163 | E | 19 | | Rq | | | So did you get it figured out? |
| tr001-164 | B | 19 | | F | | | Yeah. |
| tr001-165 | A | 19 | | F | | | They got it kind of fast |
| tr001-166 | Z | 19 | | Nc | | | Sounds of laughing |
| tr001-167 | C | 8 | | E | | Re | Group of 3 take the remainder |
| tr001-168 | B | 19 | | F | | | mhm |
| tr001-169 | E | 14 | | Rq | | | So what's the strategy from the beginning, if you have 20 tiles, what should you do first? |
| tr001-170 | D | 5 | | E | | Re | 1, and then, |
| tr001-171 | B | 11 | | Et2 | | | So that's this 1 |
| tr001-172 | B | 11 | | Et | | | Take the remainder |
| tr001-173 | A | 7 | | O | | Re | So you always do the opposite of the opponent, after you take 1 |
| tr001-174 | B | 19 | | F | | | Yeah |
| tr001-175 | A | 18 | | Et | | | Re In order to force the win |
| tr001-176 | E | 19 | | Rq | | | Does it matter who goes first |
| tr001-177 | D | 18 | | E | | Re | Team 1 wins, if they play optimally |
| tr001-178 | B | 18 | | E | | Re | Well if team 1 goes first, then they can follow the strategy to force the win |
| tr001-179 | B | 17 | | O | | T | So say that you go first, and you play incorrectly, and you take 2 |
| tr001-180 | C | 14 | 179 | Rq | 179 | Re | They were playing first and they took 2? |
| tr001-181 | B | 19 | 180 | F | 180 | | Right. |
| tr001-182 | B | 18 | 179 | El | 179 | P | Then what we can do is we can take 2, and force you to lose, if we play correctly. |
| tr001-183 | A | 19 | 182 | F | 182 | | Right. |
| tr001-184 | A | 13 | | O | | Su | So if, you played wrong, and we played right, but if we both played wrong, then it'd be back to you |
| tr001-185 | C | 19 | 184 | F | 184 | | Yeah |
| tr001-186 | B | 17 | | O | | P | Yeah so say we only played 1 |
| tr001-187 | A | 17 | 186 | Se2 | 186 | | Then you could |
| tr001-188 | B | 17 | 187 | Se2 | 187 | | Potentially |
| tr001-189 | D | 13 | 186 | O | | P | We just take 1, and we are back where we want to be |
| tr001-190 | C | 6 | | O | | R | So we just need to memorize the win numbers, or the lose numbers. |
| tr001-191 | A | 19 | 190 | F | 190 | | Right. |
| tr001-192 | D | 5,7 | 190 | C | 190 | R | Well I think it's easier to remember, take 1 and take the opposite of what your opponent does |
| tr001-193 | B | 13 | 192 | C | 190 | Ch | Well that only works for 20. |
| tr001-194 | B | 16 | 193 | Se | 193 | T | Say there was like 25 or something like that. |
| tr001-195 | B | 16 | 194 | Se | 193 | R | Then a winning number would be 22, or if you forced your opponent to 22, you win. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr001-196 | B | 16 | 194 | O | | R | So in that case it would be bad to start first |
| tr001-197 | D | 8 | 196 | A | 196 | R | Right, the tile is ever 1 mod 3. |
| tr001-198 | D | 19 | | Rq | | | Have you done? |
| tr001-199 | D | 8 | 197 | Et | 197 | Re | If the total is ever 1 mod 3 at the start of your turn, then you lose |
| tr001-200 | B | 18 | | O | 088 | Re | Right well then you can lose if they play correctly |
| tr001-201 | D | 19 | 200 | F | 200 | | Yeah |
| tr001-202 | B | 18 | 200 | Et | 088 | Re | They can force you to lose |
| tr001-203 | D | 18 | 200 | O | | | I mean I don't think it really makes sense to talk about incorrect. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr002-001 | C | 11 | | Rg | | T | So I guess we can just play a couple of games, and see while we're playing if we discover a pattern. |
| tr002-002 | C | 19 | 001 | F | 001 | | Yeah. |
| tr002-003 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-004 | C | 19 | | Nc | | | OK, so. |
| tr002-005 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-006 | C | 11 | | Rg | | | You're turn. |
| tr002-007 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-008 | C | 11 | | Rg | | | I took 2! |
| tr002-009 | Z | 19 | | Nc | | | Sounds of laughing |
| tr002-010 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-011 | C | 19 | | Nc | | | Oh, oh, oh, oh, oh! |
| tr002-012 | B | 19 | | Nc | | | Good job. |
| tr002-013 | C | 19 | | Nc | | | OK |
| tr002-014 | B | 11 | | Rg | | | I guess you guys can go first this time. |
| tr002-015 | C | 19 | 014 | F | 014 | | OK |
| tr002-016 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-017 | C | 11 | | Rg | | | It's your pick. |
| tr002-018 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-019 | C | 19 | | Nc | | | OK |
| tr002-020 | A | 19 | | Nc | | | Yeah. |
| tr002-021 | C | 11 | | Rg | | | You want to do 1 more? |
| tr002-022 | A | 19 | 021 | F | 021 | | Sure. |
| tr002-023 | B | 19 | 021 | F | 021 | | Sure. |
| tr002-024 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-025 | C | 12 | | Nc | | | I won't try to take your turn anymore |
| tr002-026 | Z | 19 | | Nc | | | Sounds of laughing |
| tr002-027 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-028 | B | 19 | | S | | | Alright, I think I see it. |
| tr002-029 | B | 9 | | O | | T | If you go first, you want to make sure your team always has an odd number. |
| tr002-030 | B | 18 | 029 | El | 029 | | To force the other team to take the last one, kind of. |
| tr002-031 | B | 19 | 030 | Et | 030 | | Cuz that's what I was trying to do, I was trying to keep it odd. |
| tr002-032 | C | 19 | 030 | F | 030 | | Yeah. |
| tr002-033 | D | 19 | | Rq | | | Seems like a good conjecture, what can we do to test it? |
| tr002-034 | D | 19 | 033 | Rp | | | I have no idea |
| tr002-035 | C | 19 | | Rg | | | Do you want to put |
| tr002-036 | D | 19 | 033 | Et | 033 | | I just naturally want to test it. |
| tr002-037 | C | 10 | | O | | T | OK so yeah, if you have, well, let's just do a smaller case. |
| tr002-038 | D | 19 | 037 | F | 037 | | mhm |
| tr002-039 | C | 2 | | O | | R | So, if you have 3, if you're picking, that's good for you because you can take 2 and win. |
| tr002-040 | A | 19 | 039 | F | 039 | | Right. |
| tr002-041 | C | 19 | | Nc | | | Um, um, and... |
| tr002-042 | B | 13 | | S | | | We have an even number... |
| tr002-043 | C | 1 | | O | | R | So for 4, so if it is 4, and you're picking, then you can't win. |
| tr002-044 | C | 18 | 043 | El | 043 | | Regardless of what you do, if you're opponent knows what they're doing. |
| tr002-045 | C | 13 | 044 | El | 044 | Rf | Because, you can take 1 or 2, and they'll still win. |
| tr002-046 | C | 19 | | Nc | | | Um. OK. |
| tr002-047 | C | 19 | | S | | | So I guess for 6, I think, so it's... |
| tr002-048 | C | 9 | | Rq | 029 | Ch | So you want it to be an odd number when you're picking, that's what you said right? |
| tr002-049 | B | 9 | 048 | E | 048 | Re | You want there to be an odd number of tiles on the board. |
| tr002-050 | C | 19 | 049 | F | 049 | | OK |
| tr002-051 | B | 17 | | Rg | | Rf | So if you take 1, |
| tr002-052 | C | 9 | | O | | R | I would take 1 to give her an even number, and then we can win. |
| tr002-053 | C | 14 | | Rq | | Ch | So there were 6 right? |
| tr002-054 | C | 19 | 053 | Et | 053 | | Yeah. |
| tr002-055 | C | 17 | | O | | R | But if you would've taken 2, then I would've had an even number, and regardless of what I took, 1 or 2, she could've made it so you guys won. |
| tr002-056 | D | 14 | | Rq | | Ch | So that's when you're talking about 4. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr002-057 | C | 14 | 056 | Rp | | | 4. |
| tr002-058 | D | 14 | 057 | F | 057 | | 4 tiles. |
| tr002-059 | C | 1 | | O | 043 | Re | If you're picking when there are 4 tiles left, you can't win. |
| tr002-060 | D | 19 | 059 | Rq | | | Why? |
| tr002-061 | C | 19 | | Nc | | | Um |
| tr002-062 | A | 7 | 060 | E | 060 | Rf | Because if they pick 1, the next team can pick 2 and leave you with the last 1, and if you pick 2, then the last team can pick 1 and leave you with the last 1. |
| tr002-063 | E | 1, 15, 10 | | Rq | | Su | 4 is bad and you know when there's 1 you're going to lose, so what about more than 4 tiles? |
| tr002-064 | C | 9 | | O | 055 | Re | I think 6 is the same way. |
| tr002-065 | C | 18 | 064 | El | 064 | | If.. all this is assuming that your opponent knows.. to not |
| tr002-066 | B | 18 | 065 | Et2 | 065 | | Mess up |
| tr002-067 | C | 19 | 066 | F | 066 | | Yeah. |
| tr002-068 | B | 17 | | O | | R | So the thing is to take 2... We could take 2... You guys could take 1, and you'd lose. |
| tr002-069 | B | 17 | 068 | El | 068 | R | If you guys took 2, we'd take 1, you guys would take 1 and you guys would lose. |
| tr002-070 | D | 18 | | O | | T | I think it always depends on the opponent, in general, generally we always assume, obviously we don't know the strategy here, but we assume the opponent knows the strategy too. |
| tr002-071 | C | 19 | 070 | F | 070 | | Yeah. |
| tr002-072 | D | 18 | | O | 070 | Re | So, I mean, once we figure out what that is, we can assume the other person won't screw up. |
| tr002-073 | C | 19 | 072 | F | 072 | | Right. |
| tr002-074 | A | 19 | 072 | F | 072 | | Yeah. |
| tr002-075 | A | 13, 1 | | O | | R | And if you're picking from 6, you should pick 2, because it would leave your opponent with 4. |
| tr002-076 | B | 14 | 075 | F | 075 | Re | OK so when there's 6 we know what to do. |
| tr002-077 | D | 19 | 076 | F | 076 | | Right. |
| tr002-078 | D | 10 | | O | | | 8. |
| tr002-079 | C | 19 | | Nc | | | So, |
| tr002-080 | B | 17 | | O | | P | You pick 2, we pick 2, this goes back to 4 |
| tr002-081 | C | 1 | 080 | F | 080 | | Back to 4 |
| tr002-082 | B | 9 | | O | | | Yeah, this team will lose. |
| tr002-083 | C | 17 | | O | | P | If I took 1, you could take, you should take 1. And if take 1, you could take 1, or if I took 2.. |
| tr002-084 | A | 17 | 083 | F | 083 | | take 2 |
| tr002-085 | C | 19 | | Nc | | | Then you guys would ... |
| tr002-086 | A | 1 | | O | 043 | Re | 4 and you lose |
| tr002-087 | C | 19 | 086 | F | 086 | | Yeah. |
| tr002-088 | C | 17 | | O | | P | But If I took 2 the first time, and you took 1, I could just take 1, and you would get that. |
| tr002-089 | C | 17 | 088 | El | 088 | | But you wouldn't want to do that. |
| tr002-090 | C | 17 | 089 | El | 089 | P | Take 1, and you would, you would want to take 1 or 2 |
| tr002-091 | B | 17 | | O | | P | Alright well, if you took 1, I would want to take 1, but then you would take 2 |
| tr002-092 | C | 19 | 091 | F | 091 | | 1 |
| tr002-093 | A | 19 | 091 | F | 091 | | 2. |
| tr002-094 | B | 13 | 091 | El | 091 | | Then you'd be screwed. |
| tr002-095 | A | 19 | 094 | F | 094 | | Yeah. |
| tr002-096 | B | 13 | 091 | El | 091 | Rf | Because we wouldn't want to take 2, because it would automatically leave you with 1. |
| tr002-097 | C | 19 | 096 | F | 096 | | Yeah |
| tr002-098 | B | 13 | | Rq | | R | So if you have 8, take 1, you'd automatically win? |
| tr002-099 | D | 19 | | Rq | | | Is someone recording this? |
| tr002-100 | D | 19 | 099 | El | 099 | | Recording the strategies for each one? |
| tr002-101 | C | 19 | 100 | Rp | | | No, Um. |
| tr002-102 | C | 19 | | O | | R | OK so, if you're picking, and there are 4, you take 1 |
| tr002-103 | B | 13 | 102 | Et2 | 075 | Re | When there's 6, you'd want to take 2. |
| tr002-104 | B | 13 | 102 | Et2 | 098 | Re | When there's 8, you'd want to take 1? |
| tr002-105 | C | 19 | 104 | F | 104 | | I mean you don't... yeah OK. And then there's... |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr002-106 | C | 13, 8 | | O | | R | Yeah, you'd take 1 at the start, because this 3 can be gotten rid of in 2 times. |
| tr002-107 | C | 7 | 106 | El | 106 | | Just by 2 people, so I would just take what you didn't take. |
| tr002-108 | C | 1 | 107 | Et | 107 | | Leaving the 4 for you. |
| tr002-109 | B | 14 | | O | | Su | 1, 2, 1, pattern so far. |
| tr002-110 | C | 19 | 109 | F | 109 | | Yeah. |
| tr002-111 | C | 17 | | O | | R | So all we have to do is get it, Well I took 2, I would've given you 8, and all you would've had to do is take 1, and you win. |
| tr002-112 | C | 17 | 111 | El | 111 | P | Because you just.. um, so I would want to take 1, and then you could, if you took 1 I would win, if you tried to take 2, I could take 2, and then you could take 1 or 2, either way, I could just do that. |
| tr002-113 | C | 17 | 112 | Et | 112 | P | Or if you had 1, I could take 1, yeah. |
| tr002-114 | B | 14 | 113 | Rq | | Ch | So for this one, you want to take 1? |
| tr002-115 | C | 19 | 114 | Rp | | | Yeah. |
| tr002-116 | C | 13 | 115 | El | 115 | Rf | Yeah because I wouldn't want to give you 8. |
| tr002-117 | C | 13 | 116 | Et | 116 | P | Yeah, take 1. |
| tr002-118 | B | 14 | | O | | Su | So it's 1, 2, 1, 1. |
| tr002-119 | E | 17 | | Rq | | | So if there's 10 and you take 1, what's your opponent going to do? |
| tr002-120 | C | 13 | | E | | P | They should take 2. |
| tr002-121 | E | 19 | | Rq | | | OK, Why? |
| tr002-122 | C | 13 | | E | | Su | Because if they take 1, they'd leave you with 8 to choose from, we've already laid out the strategy to win if you have 8, you can just take 1. |
| tr002-123 | E | 13 | | F | | Re | OK, if there's 8, you can win. |
| tr002-124 | E | 10 | | Rq | | T | If you have 9, can you win from there? |
| tr002-125 | Z | 19 | | Rg | | | Sounds of playing |
| tr002-126 | C | 13 | | O | | P | Um, you would not want to take 1, because you'd be giving me 8. |
| tr002-127 | C | 13 | | El | | P | And I would not want to give you 6, so I'd take 2, and then either way you'd lose. |
| tr002-128 | A | 14 | | C | | P | Wait you just, you take 1, |
| tr002-129 | C | 17 | | F | | P | I'd take 1 |
| tr002-130 | A | 14 | | Se | | P | You'd take 2, Because isn't it if you're left with 4 |
| tr002-131 | C | 19 | | Nc | | | Um |
| tr002-132 | A | 1 | | O | | Re | If you're left with 4 you lose. |
| tr002-133 | C | 19 | | F | | | Right |
| tr002-134 | B | 19 | | F | | | Right |
| tr002-135 | E | 6 | | Rq | | T | So you've been listing all the numbers that are OK to have in front of you and all the numbers that are not OK to have in front of you? |
| tr002-136 | E | 1 | | Et | | Re | Because you said 4 you don't want to have |
| tr002-137 | C | 19 | | F | | | Right |
| tr002-138 | E | 13 | | El | | | Because you're going to lose. |
| tr002-139 | E | 6 | | Rq | | Re | So are there other numbers that you want to avoid? |
| tr002-140 | B | 13 | | E | | Re | 8, if you take 1 you win. |
| tr002-141 | C | 14 | | Rq | | Ch | Wait can we go over 6 again? |
| tr002-142 | C | 17 | | Et | | Rf | Ok so if I'm picking first, I can take 2 or 1, |
| tr002-143 | B | 13 | | E | | Rf | You want to take 2 |
| tr002-144 | C | 13 | | Et2 | | Rf | I'd take 2 and then I'd give you 4. |
| tr002-145 | C | 14 | | El | | Re | And then with 8 it's the same kind of thing. |
| tr002-146 | C | 13 | | El | | Rf | Right, because I can take 1 and then I've got 6, and I can win by taking 2, right. |
| tr002-147 | B | 19 | | F | | | Yep. |
| tr002-148 | E | 14 | | Rq | | | Wait, is 8 OK for you to have? |
| tr002-149 | C | 19 | | Rp | | | 8... |
| tr002-150 | E | 19 | | Rq | | | Is 8 a number that you... |
| tr002-151 | E | 17 | | Rq | | | If you want to win, do you want to avoid 8? |
| tr002-152 | E | 17 | | Et | | | Or is it OK if you have 8. |
| tr002-153 | C | 13 | | E | | | No, you can't pick 2. |
| tr002-154 | C | 17 | | El | | Su | If I took 2, you'd took 1, then I could take 1, but if you took 1 I could take 2, 3, 4, so either way, depending on what you picked, I could win, right? |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr002-155 | E | 14 | | Rq | | | For which number? |
| tr002-156 | C | 14 | | Rp | | | 8. |
| tr002-157 | E | 18 | | O | | O | You want to be able to say, regardless of what you team does, can you win. |
| tr002-158 | E | 18 | | El | | | Can you guarantee a win. |
| tr002-159 | E | 19 | | El | | | You don't know what your opponent is going to do. |
| tr002-160 | E | 13 | | El | | | So you have to be able to account for either of the options. |
| tr002-161 | C | 19 | | F | | | Right, Yeah. |
| tr002-162 | E | 1 | | O | | Re | And so, at 4, you know regardless of what your opponent does, you're going to win. |
| tr002-163 | E | 13 | | El | | Re | So you want your opponent to have 4. |
| tr002-164 | E | 10 | | Rq | | T | And at 5, what would you do to make sure that your opponent loses? |
| tr002-165 | B | 13 | | Rp | | P | I'd take 1. |
| tr002-166 | E | 13 | | Et2 | | Rf | You'd take 1, because it would stick your opponent at 4. |
| tr002-167 | E | 10 | | Rq | | T | If there were 6, what would you do? |
| tr002-168 | A | 13 | | Rp | | P | Take 2. |
| tr002-169 | E | 13 | | Et2 | | Rf | Take 2, because that again sticks them at 4. |
| tr002-170 | E | 10 | | O | | T | So now, we're up to 7. |
| tr002-171 | E | 19 | | Rq | | | What's going to happen? |
| tr002-172 | E | 15 | | El | | O | You have the option of taking 1 or 2. |
| tr002-173 | C | 19 | | F | | | mhm |
| tr002-174 | E | 17 | | Se2 | | P | If you take 1 |
| tr002-175 | A | 17 | | E | | P | You'd want to take 2 second, because then |
| tr002-176 | C | 17 | | Et2 | | P | Because if you took 1 they could |
| tr002-177 | A | 17, | 1 | Et | | P | They could take 2, which would leave you from the 4 to choose from, which means you lose. |
| tr002-178 | E | 19 | | F | | | Right. |
| tr002-179 | E | 17 | | Rq | | P | So what if you took 2 from 7. |
| tr002-180 | E | 19 | | Et | | | What would they do? |
| tr002-181 | C | 13 | | Rp | | P | They'd take 1. |
| tr002-182 | B | 13 | | O | | R | So you don't want to end up at 7. |
| tr002-183 | D | 6 | | Et2 | | Re | 7 is another number to avoid. |
| tr002-184 | E | 10 | | Rq | | T | You're avoiding 7, what should you do at 8? |
| tr002-185 | C | 13 | | E | | R | I want to take 1, to leave them with 7. |
| tr002-186 | E | 13 | | F | | | To give them 7. |
| tr002-187 | C | 19 | | F | | | Right. |
| tr002-188 | C | 13,10 | | Se | | T | And then so 8 you can win, add a ninth one, you want to take |
| tr002-189 | A | 13 | | Et2 | | | 2 |
| tr002-190 | C | 13 | | O | | P | 2 to give them 7 |
| tr002-191 | C | 19 | | Rq | | Ch | But, so, OK, so yeah, if, and then for 10, and we're sure about 8? |
| tr002-192 | C | 13 | | El | | P | Take 1 and give them 7. |
| tr002-193 | A | 19 | | F | | | Yeah |
| tr002-194 | B | 13 | | O | | R | So for 9 you want to take 2, to give them 7. |
| tr002-195 | C | 19 | | F | | | mhm |
| tr002-196 | E | 10 | | Rq | | T | And what about 10? |
| tr002-197 | B | 7 | | E | | P | If I take 2, and you guys take 1, you lose. |
| tr002-198 | B | 13 | | El | | P | So I'd want to take 1, |
| tr002-199 | C | 17 | | Et2 | | P | And I would take 2 |
| tr002-200 | A | 6 | | El | | R | And you'd lose, so 10 is another one. |
| tr002-201 | B | 13 | | O | | R | Then 11, just take 1. |
| tr002-202 | B | 13 | | El | | R | 12 take 2. |
| tr002-203 | A | 13 | | F | | R | 12 take 2. |
| tr002-204 | A | 6, 13 | | O | | R | 13, another one you want to avoid. |
| tr002-205 | B | 19 | | F | | | Yeah. |
| tr002-206 | B | 13 | | O | | R | 14, take 1, 15 take 2. |
| tr002-207 | B | 6, 13 | | El | | R | 16 you want to avoid. |
| tr002-208 | E | 6 | | Rq | | | So what's the next one to avoid? |
| tr002-209 | B | 6 | | Rp | | R | 19. |
| tr002-210 | E | 19 | | F | | | Good. |
| tr002-211 | E | 15 | | Rq | | S | So you're starting with 20 tiles, what should be your first move? |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr002-212 | B | 13 | | Rp | | P | You want to take 2. |
| tr002-213 | E | 19 | | Nc | | | Well now... |
| tr002-214 | C | 13 | | O | | P | My team wants to avoid, so we'd take 1. |
| tr002-215 | B | 19 | | F | | | Yeah. |
| tr002-216 | E | 3 | | Et2 | | T | Force the other team to have the 19. |
| tr002-217 | E | 10 | | Rq | | T | And how can, if they'...re at 19, how can you make sure they get down to 16? |
| tr002-218 | A | 6, 7 | | E | | R | Keep forcing them, to the numbers, because then regardless, if they take 1 you take 2, if they take 2 you take 1. |
| tr002-219 | E | 19 | | F | | | Good. Ok. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr003-001 | C | 11 | | Rg | | | OK, so who wants to, should we go first? |
| tr003-002 | B | 19 | 001 | F | 001 | | Sure |
| tr003-003 | Z | 19 | | Rg | | | Sounds of playing |
| tr003-004 | C | 11 | | Rg | | | Go for it, we don't know the strategy |
| tr003-005 | C | 19 | | Nc | | | Oh no, I gave bad advice |
| tr003-006 | A | 19 | | Nc | | | No, either way ...yeah |
| tr003-007 | B | 1 | | O | | | When it gets down to 4// |
| tr003-008 | A | 13 | 007 | Et2 | 007 | R | Once it gets down to 3 or 4 there's only two moves left |
| tr003-009 | B | 14 | 008 | Rq | | | Yeah, so there's 4 left right? |
| tr003-010 | C | 19 | 009 | F | 009 | | Right |
| tr003-011 | B | 13 | 010 | Rg | 008 | Ch | And I took 1, and then you took 2, and then I had to take the Poison Tile |
| tr003-012 | C | 13 | 011 | Et2 | 011 | Rf | So even if you would've taken 2, she would've taken |
| tr003-013 | B | 13 | 010 | Et2 | 011 | Rf | taken 1, and I still would've taken the poison tile |
| tr003-014 | C | 1 | | O | | R | Ok, so when there was 4 left, the person who was next to go, loses |
| tr003-015 | A | 19 | 014 | F | 014 | | Yeah |
| tr003-016 | B | 19 | 014 | F | 014 | | Yeah |
| tr003-017 | C | 19 | 014 | F | 014 | | Yeah |
| tr003-018 | B | 13 | 014 | Et2 | 014 | | The person whose turn it is to pick |
| tr003-019 | C | 11 | | Rg | | | Ok, Play again |
| tr003-020 | A | 11 | 019 | Rg | | | I'll go first this time |
| tr003-021 | B | 19 | 020 | F | 020 | | OK |
| tr003-022 | Z | 19 | | Rg | | | Sounds of playing |
| tr003-023 | C | 11 | | Rg | | Ch | Wait, sorry what did you just take? |
| tr003-024 | B | 11 | 023 | Rg | | Ch | I took 1 |
| tr003-025 | C | 11 | 024 | Rg | | Ch | You took 1 |
| tr003-026 | B | 11 | 025 | Rg | | Ch | And Lisa just took 1 |
| tr003-027 | C | 11 | 026 | F | 026 | | Ok, got it. |
| tr003-028 | Z | 19 | | Rg | | | Sounds of playing |
| tr003-029 | A | 19 | | Nc | | | Darn |
| tr003-030 | C | 11 | | Rg | | | OK, so you lose |
| tr003-031 | A | 14 | | Rg | | Ch | What did you do last time? |
| tr003-032 | A | 14 | 032 | Et | 032 | Ch | You took 2? |
| tr003-033 | B | 19 | 032 | F | 032 | | Yeah, yeah. |
| tr003-034 | C | 10 | | O | | T | OK, so this time for the game, put 5 left. |
| tr003-035 | B | 11 | | Rq | | P | So someone take 2, right now to get down to 3? |
| tr003-036 | A | 17 | 035 | E | 035 | P | Well it could be 2 2 and 1, or it could be 1 2 and 2, either way there's only |
| tr003-037 | C | 19 | | Nc | | | But whoever, like, but |
| tr003-038 | A | 17 | 036 | El | 036 | | It could be 2 1 1 1 |
| tr003-039 | C | 13 | | Rq | | Ch | 5, if it was your turn, you'd want to take 1 right? |
| tr003-040 | C | 1 | 039 | E | 039 | Re | To get it down to 4 |
| tr003-041 | A | 19 | 040 | F | 040 | | Yeah |
| tr003-042 | B | 19 | 040 | F | 040 | | Yeah |
| tr003-043 | C | 13 | 040 | El | 040 | Re | So that the other person would lose |
| tr003-044 | B | 17 | 043 | Et2 | 043 | Re | Yeah, so the next person would take 2 or 1 |
| tr003-045 | C | 13 | | O | | R | Ok so what if there was, so if there was 6 left, you'd want to take 2 |
| tr003-046 | A | 1 | 046 | A | 045 | Re | Yeah cause either way you want there to be 4 left, so you don't want there to be 5 when it's. |
| tr003-047 | A | 2 | | O | | Re | If there's 3 left, and it's not your turn, you should lose, because the other person should take 2. |
| tr003-048 | C | 10 | | O | | T | OK so let's look at 7, because 4, 5, and 6 we know. |
| tr003-049 | B | 10 | 048 | F | 048 | | 7 |
| tr003-050 | C | 19 | 048 | Rq | | | What would be our strategy? |
| tr003-051 | A | 1 | 050 | Rq | 048 | P | So if we want to leave the other person with 4? |
| tr003-052 | A | 19 | 051 | Se | 051 | P | Probably best to take... |
| tr003-053 | A | 13 | 051 | Se | 051 | | Either way they can leave you with |
| tr003-054 | C | 13 | 051 | O | | R | so if there's 7 you lose |
| tr003-055 | B | 19 | 054 | F | 054 | | Yeah |
| tr003-056 | A | 18 | 055 | Et2 | 055 | | If the other person knows |
| tr003-057 | B | 19 | | Nc | | | Um |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr003-058 | C | 18 | 056 | O | | | Well they should know the strategy |
| tr003-059 | B | 19 | 058 | F | 058 | Yeah | |
| tr003-060 | C | 18 | 058 | Et | 058 | | If you play the game right |
| tr003-061 | A | 18 | 058 | Et2 | 058 | | If you figure out the pattern |
| tr003-062 | B | 19 | 061 | F | 061 | | Yeah |
| tr003-063 | C | 10 | | O | | T | So yeah, OK, what if there's 8. |
| tr003-064 | C | 13 | 063 | El | 063 | R | Then you'd want to take 1 |
| tr003-065 | A | 13 | 064 | Et2 | 064 | Re | So that they're left with 7. |
| tr003-066 | C | 19 | 065 | F | 065 | | Yeah |
| tr003-067 | A | 19 | 065 | F | 065 | | Yeah |
| tr003-068 | B | 19 | 065 | F | 065 | | Yeah |
| tr003-069 | A | 6 | | O | | Su | OK, so if you were up and there are either 7 or 4 left, you lose. |
| tr003-070 | C | 19 | 069 | F | 065 | | Right, OK |
| tr003-071 | A | 13 | | O | | R | So if there's 8 or 9 left, and it's your turn, you want to take 2 or 1 so that it leaves them with 7. |
| tr003-072 | C | 13 | 071 | A | 071 | Rf | So it's to 7. OK |
| tr003-073 | B | 6 | | O | | So | 7 and 4 are pretty critical |
| tr003-074 | C | 19 | 073 | F | 073 | | OK |
| tr003-075 | B | 10 | | Rq | | Ch | What'd we, this is 8 right? |
| tr003-076 | C | 13 | | O | 075 | Ch | Oh, but if it's 8, you want to take 1, so that the other person's at 7 |
| tr003-077 | A | 13 | | O | | R | And if its 9, you want to take 2. |
| tr003-078 | A | 10 | | Rq | | | If there's 10 |
| tr003-079 | C | 10 | 078 | F | 078 | T | So if it's 10 |
| tr003-080 | A | 13 | 078 | El | 079 | P | If you pick 1 or 2 then the other person has the 9 or 8 to make it 7. |
| tr003-081 | A | 13 | 078 | El | 080 | R | So 10 you lose. |
| tr003-082 | A | 8 | 078 | El | 078 | R | So it seems to be going up by 3s. |
| tr003-083 | C | 19 | 082 | F | 082 | | Um, yeah. So then, yeah, |
| tr003-084 | C | 13 | | O | | R | So 11 or 12 you'd be able to get them. |
| tr003-085 | C | 13 | | Rq | | Ch | So then what about at 13, 13 you lose? |
| tr003-086 | A | 13 | 085 | A | 085 | R | 13 you lose. |
| tr003-087 | A | 17 | | Se | | | Oh so, if you go first you just want to take |
| tr003-088 | C | 19 | | Rq | | | How many tiles... |
| tr003-089 | A | 5,18 | 087 | O | 087 | R | There's 20, so if you go first, you just want to take 1 tile, so the other person already loses |
| tr003-090 | B | 19 | 089 | F | 089 | | Yeah |
| tr003-091 | A | 7 | 089 | El | 089 | R | As long as you do the opposite of them |
| tr003-092 | B | 6 | | O | 089 | Ch | 13, 16, 19 |
| tr003-093 | C | 19 | 092 | F | 092 | | Yeah, |
| tr003-094 | A | 4,18 | | O | 089 | Re | So whoever goes first, wins, if they play it smart |
| tr003-095 | C | 18 | | Rg | | O | So we're trying to figure out, we just want to know how to play to make the other person lose |
| tr003-096 | A | 17 | | Rq | | T | But if you don't go first, how can you win? |
| tr003-097 | A | 18 | 096 | Et | 096 | P | Hope the other person doesn't know what they're doing |
| tr003-098 | B | 18 | 096 | E | 096 | P | Make a mistake. |
| tr003-099 | C | 19 | 098 | F | 098 | | Yeah |
| tr003-100 | A | 11 | | Rg | | Ch | Alright let's make sure that this works. |
| tr003-101 | C | 19 | 100 | F | 100 | | OK, Yeah. |
| tr003-102 | C | 4 | | O | 100 | Su | OK so based off what you know now, whoever goes first should win. |
| tr003-103 | A | 11 | | Rg | | | Do you want to go first? |
| tr003-104 | B | 11 | 103 | Rp | | | OK, sure, I'll probably screw it up |
| tr003-105 | C | 11 | | Rg | | | So you, so now it's down to 17 |
| tr003-106 | B | 11 | 105 | F | 105 | | 17 |
| tr003-107 | C | 11 | | Rq | | | Do you remember which one is the one you want to get it to? |
| tr003-108 | A | 19 | 107 | F | 107 | | Yeah |
| tr003-109 | B | 6 | 107 | E | 107 | | I want to get it to 16 |
| tr003-110 | C | 19 | 109 | F | 109 | | Yeah |
| tr003-111 | Z | 19 | | Rg | | | Sounds of playing |
| tr003-112 | A | 7 | | O | 091 | Re | Basically you want to take the opposite of what they're doing. |

| SentenceTag | Speaker | Bit | CT | C | PT | P | Sentence |
|---|---|---|---|---|---|---|---|
| tr003-113 | A | 7 | 113 | E | 113 | | If they take 1, you take 2, if they take 2 you take 1. |
| tr003-114 | A | 8 | 113 | El | 113 | | It's always going down in 3s each time |
| tr003-115 | B | 19 | 115 | F | 115 | | Right |
| tr003-116 | C | 19 | 115 | F | 115 | | Yeah |
| tr003-117 | B | 11 | | Rg | | | I'll take 1 |
| tr003-118 | E | 18 | | Rq | | | So what's the strategy? |
| tr003-119 | C | 5,1,6 | | E | | Re | If you go first, you want to take 1, because starting at 4, if there's 4 tiles left, you automatically lose no matter what you play, and that happens at, 4, 7, 10, 13, 16, and 19 tiles. |
| tr003-120 | C | 6 | | El | | Re | You always want to get the other player at those marks. |
| tr003-121 | C | 13 | | El | | Re | And then if you don't go first, hope that the other person... |
| tr003-122 | A | 13 | | Et2 | | | Doesn't know that |
| tr003-123 | Z | 19 | | Nc | | | Sounds of laughing |
| tr003-124 | E | 19 | | F | | | Right, OK |
| tr003-125 | E | 14 | | Rq | | | Rf Can you give some justification about why this is going to work? |
| tr003-126 | E | 6 | | Rq | | Rf | Why are 1, 4, and 7 the numbers that you want to avoid? |
| tr003-127 | B | 19 | | Nc | | | Yeah so 1 |
| tr003-128 | A | 4,8 | | E | | Rf | Because whoever goes first always has control, basically because if the other player takes 1, you can always make it a 3-some within every two turns. |
| tr003-129 | A | 7 | | Et | | Rf | If they take 2 you take 1. |
| tr003-130 | A | 6,8 | | Et | | Rf | So it always adds up to 3, so it always ends up on these numbers. |
| tr003-131 | C | 1 | | O | | | But I think she wants to know, about 4, well I mean we talked about the 4, because no matter if you take 1 or 2, the other person can.. |
| tr003-132 | A | 1 | 131 | Et2 | 131 | | Can leave you 1 |
| tr003-133 | C | 19 | 132 | F | 132 | | OK |
| tr003-134 | B | 6 | | O | | T | I wonder if she's looking for a formula, or why are those numbers the way they are. |
| tr003-135 | B | 17 | | O | | T | I wonder if it has to do with the total number of tiles. |
| tr003-136 | B | 16 | | O | | T | Or like if we had 30 tiles, it'd still be those numbers |
| tr003-137 | A | 19 | 136 | F | 136 | | Yeah |
| tr003-138 | C | 19 | 136 | F | 136 | | Yeah |
| tr003-139 | A | 16 | | O | | T | It'd change a lot if you could choose like other 0 or 3 tiles instead of just 1 or 2. |