Question Generation: Papers from the 2011 AAAI Fall Symposium (FS-11-04)

A Graph Theory Approach for Generating Multiple Choice Exams

Sarah K. K. Luger

Institute for Language, Cognition and Computation
The University of Edinburgh
Edinburgh, UK
{s k.k.luger@sms.ed.ac.uk}

Abstract

It is costly and time consuming to develop Multiple Choice Questions (MCQ) by hand. Using web-based resources to automate components of MCQ development would greatly benefit the education community through reducing reduplication of effort. Similar to many areas of Natural Language Processing (NLP), human-judged data is needed to train automated systems, but the majority of such data is proprietary. We present a graph-based representation for gathering training data from existing, web-based resources that increases access to such data and better directs the development of good questions.

1 Introduction

Systems that automate judging question difficulty have value for both educators, who spend large amounts of time creating novel questions, and students, who spend a great deal of time taking tests. The current approach for measuring question difficulty relies on inspecting exam results and looking at the answer distracters picked most often by high-scoring students in comparison to those chosen by low-scoring students. This method relies on models of how good pupils will perform and contrasts that with their lower-performing peers. Inverting this process and allowing educators to test their questions before students answer them would speed up question development and utility.

In this paper we consider only Multiple Choice Questions, (MCQ). A question or "item" consists of several parts. The stem, or "question statement", presents a query that is best answered by one of the answer options. The answer options, or "answer alternates", will include the *answer* and the *distracters*.

In this paper we present an alternative method for building exams from sets of questions that have been answered by students. As exams are crucial for analyzing the difficulty of questions, building them from non-exam data aids in automating the generation of MCQ.

2 Test Item Difficulty and Item Analysis

A question may be difficult in many ways. The stem may be confusing or poorly scoped. The topic of the stem may be from an obscure corner of a discipline or use ambiguous terminology. Further, when a question has multiple answer options, high quality, incorrect options make a question difficult.

To measure question difficulty, researchers have devised a method for judging both the difficulty of the question and the differentiation power of the answer options (*Item Analysis)*. Once a cohort (for this example, 100 students) has taken a test containing suitable questions, the process for ascertaining this information is as follows [Gronlund, 1981]:

1) The exams are graded and ranked from highest score to lowest.

2) The set of 100 students is split into three groups that represent the top-scoring, middle-scoring, and lowest-scoring students. These three groups are commonly split, lower 27%-middle 46%-upper 27%.

3) The middle set of (46) exams is excluded.

4) For each test item (question), the number of students in the upper and lower groups who chose each answer option is tabulated in a template. Table 1 illustrates a sample filled-in template, including all omissions.

5) *Item Difficulty* is measured by the percentage of students who answered a question correctly. The lower the percentage, the more difficult the question is. In Table 1, the correct answer is **B** and the question has an item difficulty of 35%, as shown in column 6.

6) *Item Discriminating Power* is the difference between the number of high-scoring students versus the number of low-scoring students who chose the same answer option. It is an indicator of item difficulty on an answer option-by-answer option basis. This is shown in Table 1, column 7 and in more detail in Figure 1.

Column 1 Item Number	Column 2 Choice Letter	Column 3 # in High-Scoring 27%	Column 4 # in Middle-Scoring 46%	Column 5 # in Low-Scoring 27%	Column 6 Total Number	Column 7 Difference Between #'s in Col. 3 and 5
1	A	4	14	5	23	-1
	B	11	18	6	**35**	5
	C	3	3	2	8	1
	D	3	5	3	11	0
	E	5	2	9	16	-4
	OMIT	1	4	2	7	0
	TOTAL	27	46	27	**100**	0

Table 1 Item Analysis examines the answer and distracter choices groups of students made for a single question.

Distracters and their respective discriminating power values are shown in Figure 1. A "good" or difficult distracter is one that catches or distracts more good students than bad students; such items have a positive number in column 7.

This method for judging question difficulty and item discriminating power relies on models of student performance from the three groups previously mentioned. Comprehension and aptitude tests seek to present questions that can be correctly answered by students who understand the subject matter and to confuse all other students with seemingly viable alternate answer options (distracters).

A high-scoring student is one who answers most questions correctly, but when his or her answers are incorrect, chooses the best distracters. A low-scoring student will choose any of the answer options seemingly at random. A difficult question is one whose answer options are all deemed viable to a high-scoring student. That cohort will behave like low-scoring students, with a near equal spread of multiple distracters being chosen.

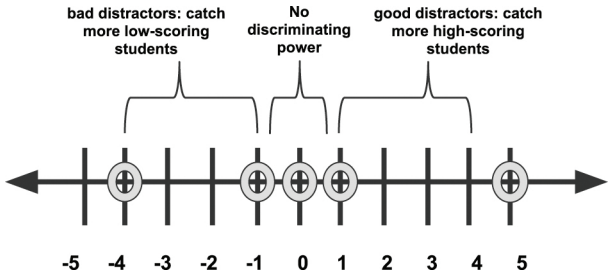

Figure 1 Item Discriminating Power as a spectrum of distracter classes.

We measure the relationship between the question and the answer option in a way that mirrors student performance as ascertained by Item Analysis. Our tool could be run prior to giving exams and would eliminate --or at least reduce-- the feedback loop of student results that directs future question development. This would help avoid using questions that do not produce the desired discriminating effect.

3 Language and Question Difficulty

Many NLP-based exam generation systems rely heavily on previously produced real exam data to refine how similar distracters need to be to the correct answer to be "good" [Mitkov et al., 2009]. In the case of standardized comprehension or aptitude exams, this means having access to sets of exam data, which include the questions and detailed, question-by-question results from thousands of students. Unfortunately, such ideal data is very difficult to obtain.

We procured data for two sets of MCQs from University-level introductory biology classes using the PeerWise[1] question creation system [Denny, 2009]. PeerWise is a free, web-based, question repository that allows classes to set up shared environments where students create questions that are subsequently used by their peers as a study aid. Instructors can review the questions or use some of the better questions for future exams. The repository consists of questions that are created by students and answered by their classmates. Since answering these questions may not be compulsory, the resulting data is a set of questions that have been answered by students but, not all of the questions have been answered by the *same* students.

Developing a collaborative exam-building environment, such as PeerWise, has resulted in more than just a set of potential exam questions associated with the related curriculum. The students self-police the quality and correctness of questions via a ratings system and comments section. Question difficulty is ranked from 1 to 3, 3 being the most difficult and question quality is measured from 0 to 5, 5 being the highest. These question ratings are saved and can be used as a comparison to how the students actually perform on the questions. Further, creating questions forces students to understand concepts adequately enough not only to make a correct statement, but also to find good distracters that in-

[1] http://peerwise.cs.auckland.ac nz/

deed distract their classmates. In classes that use PeerWise, instructors may make creating questions a voluntary, mandatory, or graded component of their class.

In PeerWise, there are three steps for authoring and answering questions. The first is to write, or author the question, using the provided template that allows a window for typing the query and input cells for up to 5 distracters, A-E. Then, there is space for a student to add an explanation of the question and refer to the related textbook or course notes via tick boxes. These suggestions are based on previous links made between other questions and the course material. There is also input space for describing why the correct answer is the best answer choice. This is especially useful in cases where the answer options are closely related topics. Students may contribute multiple questions and all questions that they create are linked via a unique, but publically anonymous ID.

The second component of the web tool allows the students to have a test-like experience by answering questions. The answer screen looks like that of a conventional online test, with tick boxes associated with each possible answer. An example of the PeerWise biology data is question 31522:

What is the name of the areas between osteons?

A) canaliculi
B) lacunae
C) lamellae
D) interstitial lamellae (correct answer)
E) Volkmann's canals

When a student is presented with this question, a statement at the top of the page notes how many people have previously answered the question (in this case 245 other classmates) and what was the average quality rating they gave the question (in this case 2.77). Not all of the students rate all of the questions and in this instance the ratings are based on 62 responses.

The third step is to compare your answer choice to the correct one and then to rate the question. For question 31522, the correct answer, D, was chosen 87 times, or by 35% of the students. The number of students who chose each answer option is listed as well as the question explanation. Students are then given the chance to rate the question both in terms of overall quality and in regards to difficulty. Finally, comments, suggests and edits may be included and these are emailed to the student who authored the question so that any flagged errors may be corrected. The format of the comment section is similar to that of an electronic bulletin board, so it allows the classmates to discuss the questions back and forth.

The process of choosing questions for the datasets consisted of automatically collecting the subset of questions that used inverse definition constructions such as "is called", "known as", and "is named" via regular expressions. Inverse definition questions describe a term or process by providing a definition and seek the name of the process. This question format is frequently used in the sciences where mastering domain-specific concepts are a key measure of comprehension.

Further filtering of the questions removed any questions that contained or were structured with images, symbols, true-false, analogies, or negation. Questions were also omitted that used "often" or "usually", fill-in the blank format, or required set membership. In addition, only questions with 4 or 5 distracters were used. This pre-processing reduced the data Set1 of 752 biology questions to 148.

Then, the question sets were manually reviewed and all related questions materials were collected for processing. These materials consisted of the unique question ID, the timestamp of when the question was taken, the unique student ID, the average rating, (0 to 5), the average difficulty, (1 to 3), the total number of responses, the total number of ratings, the correct answer, the number of answer options, the text of the question, the text of the answer options, and an explanation, if present. All of the question materials were provided in plaintext.

4 The Adjacency Matrix Approach

Since Item Analysis depends on splitting the group of students who took the test into three subgroups, we need the scores and student set size to be sufficiently large. Our sample data has many omissions, as students choose which questions they want to try answering.

Our approach for representing the individual student question answering relationship is with a graph: an "exam", where every student answers every question would be a complete bipartite graph (or biclique) [Bondy and Murty, 1976]. We are seeking a good set that is similar to an exam. By using a heat map, where correlated data appears as darkened images, (Figure 3) to show the group of students who have answered the same questions, we are presented with a realistic exam where there are a few holes, omitted questions. These would be missing edges in Figure 2. The heat map presented in Figure 6 shows the data sorted to reveal the most dense group of students who have answered the same question. It also allows further analysis of this dense region to discover a maximal graph, or exams with no omissions.

Finding a biclique in a larger semi-definite correlation matrix is an NP-complete problem [Aho, Hopecroft, and Ullman, 1974]. Discovering the single maximal clique is the ideal scenario, but in this situation, we only need to find a sufficiently large clique. Seeking the set of students who have answered the same questions would mean comparing each student's questions to the questions answered by all other students, pairwise, iteratively. That is an NP-hard problem.

Given an incidence matrix M of students and questions, where the rows of M correspond to students and the columns correspond to questions, we can generate covariance matrices S and Q. S is defined as $M \times M^{T}$ which

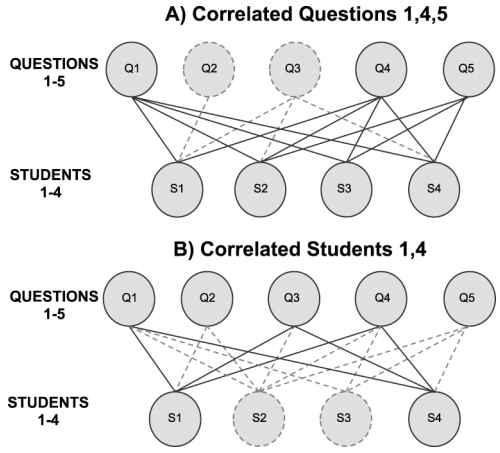

Figure 2 Correlated Questions and Students as connected cliques in a bipartite sub-graph. The edges represent each unique question-student pair that is recorded every time a student answers a question. In the top graph the solid edges belong to the most correlated questions and, in the bottom one, belong to the most correlated students.

generates a covariance matrix where S_{ij} shows how many students questions student i has answered in common with student j. Q is defined as M^{T} x M which generates a covariance matrix where Q_{ij} shows how many students have answered question i as well as question j. This can be seen graphically in Figure 2. S and Q can then be used heuristically to compute a sufficiently large clique of questions that have all been answered by the same set of students.

The steps for building and sorting the covariance matrices are as follows:

1) Collect the data in triples of student ID, question ID, and answer choice.

2) The students are ordered by the number of questions they answered.

3) Build the incidence matrix M, with students corresponding to rows and the questions to columns. If a student answered a question, a 1 is placed in the appropriate column, if they did not, a 0 is placed in the space. The incidence matrix in Figure 5(1) is the bipartite graph shown in Figure 2.

4) Compute $S = M$ x M^{T}. A heat map of S can be seen in Figure 3.

5) Compute $Q = M^{\mathrm{T}}$ x M.

6) We can find the most correlated students by computing the vector s by summing over the rows of S. Thus, $s = \Sigma_i S_{ij}$. We can then sort the rows and columns of S based on the ordering of $\mathbf{s}$ as S is symmetric. This effect can be seen in Figure 6.

7) As above, we can find the most correlated students by computing the vector $\mathbf{q} = \Sigma_i S_{ij}$. We can then sort the rows and columns of Q based on the ordering of $\mathbf{q}$.

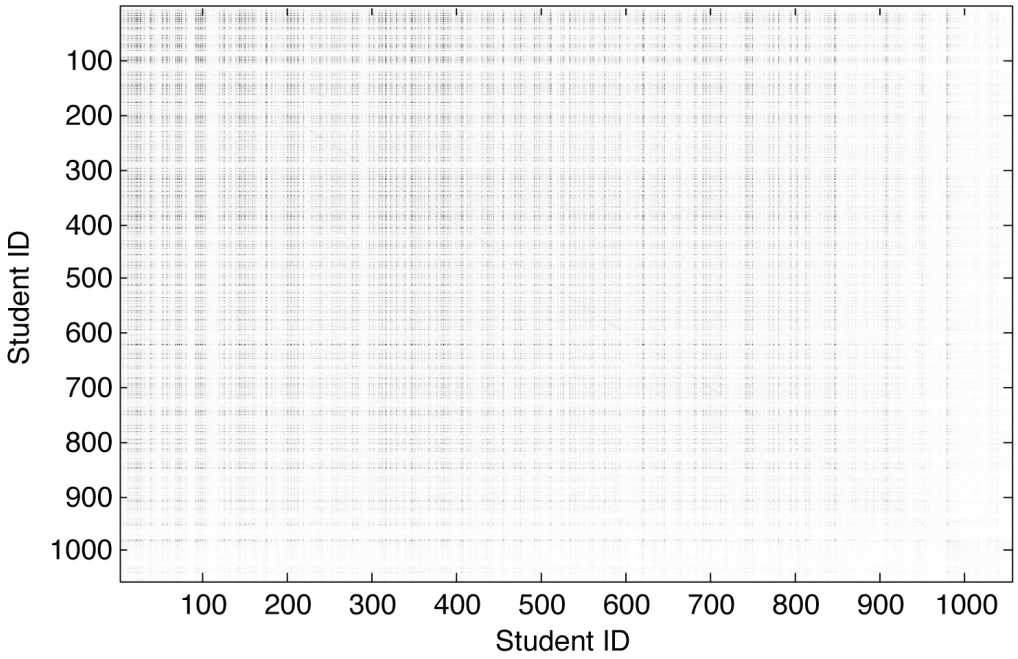

Figure 3 Heat map of the correlated students before they are sorted to reflect the most correlated sets. Here, the white represents uncorrelated pairs and the black shows correlated pairs.

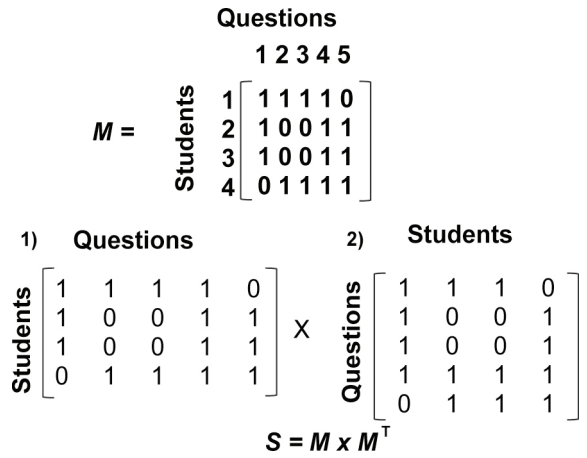

Figure 4 Setting up the Correlation Matrix.

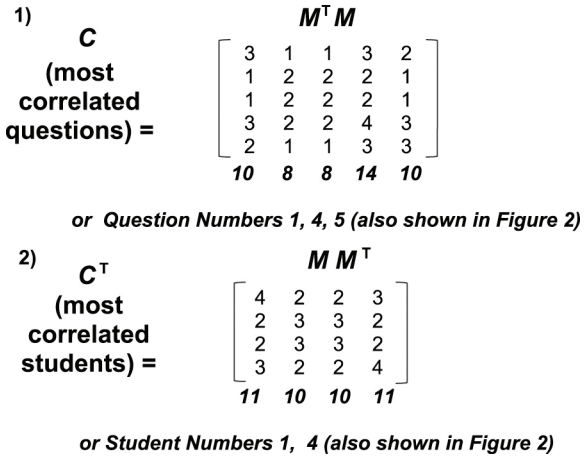

Figure 5 In the Correlation Matrix, using transpose and sum to reveal the most associated questions and students.

This sorting process provides a sound heuristic for selecting highly correlated students and questions. We use a heat map (Figure 6) to show the most dense group of students who have answered the same questions using the aforementioned methodology. This presents a realistic exam where there are a few holes, i.e. omitted questions. Again, these would be missing edges in Figure 2. The heat map also allows further analysis of this dense region to discover a maximal graph, or exams with no omissions.

In the example shown in Figures 4 and 5, each question was given an identifier from 1 to 5. Each student, of which there were 4, was given an identifier from 1 to 4. An incidence matrix M of size 5 x 4 was generated in which each row corresponds to a student and each column to a question. If a student answered a question, a 1 was entered into the incidence matrix at the appropriate row and column. All of the other spaces contained 0s.

Given M might look like Figure 4(1), the transpose of M, M^T, might look like Figure 4(2). Multiplying the matrix M^T with M will then produce a covariance matrix C of 5 x 5, whose sum reveals the most correlated questions. Each cell of the covariance matrix contains the "correlation index" C_{ij} that is a metric of how well correlated sentence i is with sentence j. This is shown in Figure 5(1). The sum of M x M^T would be of size 4 x 4 and present the most correlated students, Figure 5(2).

This graph-based algorithm, at its essence, prioritizes the set of students and questions that should be searched first to create the optimal, desired exam. A user may seek multiple sets of exams with a varying balance between the number of students and the number of questions. For example, one may seek an exam with a large number of students and a small number of questions, in an effort to give a test that quickly differentiates students into performance cohorts via item analysis.

5 Results and Analysis

The bipartite sub-clique approach was implemented on two sets of PeerWise class data consisting of a set of questions, some of which had been answered by some of the students in the class. Set1 consisted of 148 questions and 1055 students. There were 28,049 edges between the questions and students (similar to Figure 2). Set2 consisted of 134 questions and 900 students. There were 31,765 edges, or question-student pairings.

A maximal clique for Set1, that mirrors the 100-student exam example shown in Section 2, would consist of 100 students who have answered the same 13 questions. For Set2, this same exam model produces a set of 100 students who have answered the same 88 questions. This was determined using the sum process described Figure 5(2).

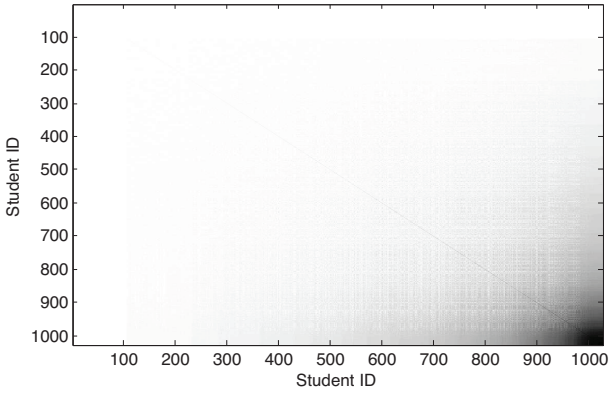

Figure 6 Heat map of the covariance matrix for data Set1, based on the number of students who answered the same questions. The x-axis orders the students by who answered the most questions multiplied by those students transpose. The y-axis, is those students transpose. Again, white represents uncorrelated pairs, whereas black represents correlated pairs.

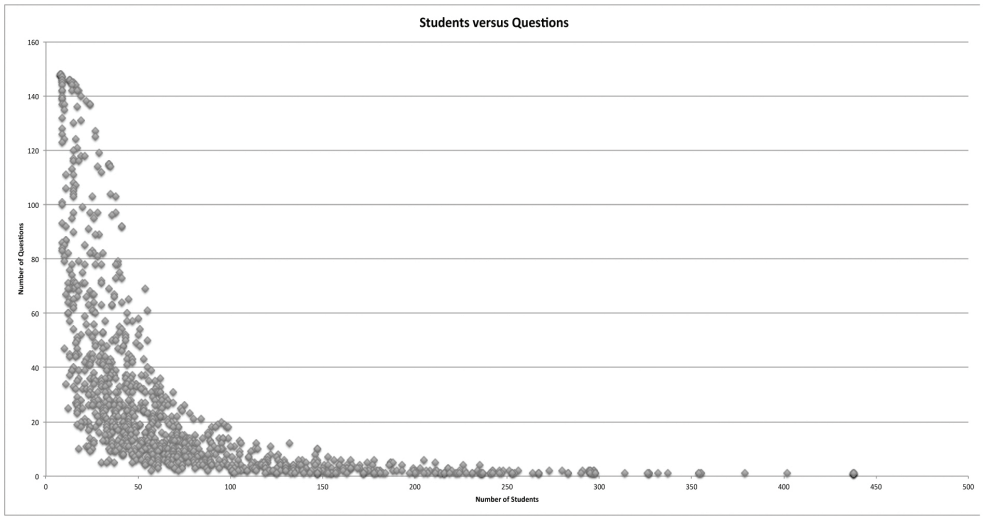

Figure 7 A set of results for data Set1, which shows a set of potential exams. Each point represents a unique virtual exam for a given number of students and a given number of questions. The x-axis represents the number of students and the y-axis the number of questions.

6 Future Work

The primary focus of future work is the identification of the most statistically significant balance between the numbers of students and the number of questions in an exam size. To build our exams we are seeking two maximums: the most number of students answering the largest number of the same questions. Thus, there is not one answer, but a set of possible answers that may reveal this ideal balance. For example in Figure 7, the adjacency matrix approach produced multiple exams ranging in size from one question answered by the same 438 students to 148 questions answered by the same 9 students. We can then apply item analysis to the resulting exams.

There are two initial concerns that need to be addressed in regards to the new, created exams. One potential problem is the hypothesis that PeerWise attracts the better performing students to practice and build their expertise in a field. The better students may tend to both author and answer more questions than their lower-performing peers. Thus, the PeerWise system may skew Item Analysis from a more conventional bell-curve across performance cohorts to a tight cluster of top-scoring students versus a long tail of the middle and lowest performing students. This hypothesis is currently being tested. So far, there is strong evidence that a highly clustered subset of the total set of students tend to answer a close grouping of questions.

The second concern regarding the format of these exams is that the adjacency matrix-based exam creation is simply a data pre-processing step that provides potential evaluation materials in a faster and cheaper manner than other options. Once the exams have been created, the next step is to grade them and begin sorting the students into cohorts based on their performance. An underlying issue is whether the questions themselves are discriminating. Are there thresholds that should be met for a question to be deemed *good enough* to split the top-performing students from the lower-performing ones?

A naïve approach might be to force a threshold of question difficulty onto the exams. An example of question difficulty is shown earlier in the paper and computed in column six of Table 1. In this example the correct answer is **B** and the question has an item difficulty of 35%. The lower the percentage, the more difficult the question is. Perhaps exam questions need to have a difficulty of 75% to be discriminating for deeper Item Analysis. In one experiment 85% difficulty was the threshold for question inclusion, and only questions below 85% were added in exam building. Again, the lower the percentage the more difficult the question, so the excluded questions were relatively easy. This reduced the upper bound on the question-student exam size from 148 questions answered by the same 9 students to 144 questions answered by 9 students. Additional experiments that iterate through lower difficulty thresholds and compare how the resulting exams effectively split students into meaningful cohorts are being currently performed.

Clearly, it is not simply the difficult questions that do not discriminate as questions that are too easy also fail to effectively group students based on their performance level. These might be identified by rules where questions in which the same answer option is chosen by more than 75% of students or where only 2 of the 5 answer alternates are chosen by any of the students, are omitted from the test set. Continued work is needed to discover the best constraints for discovering discriminating questions.

One final future improvement would allow exams to contain students who omit questions, as they do in natural test taking. Discovering what percentage of questions may be omitted in an exam without negatively affecting the statistical significance of the cohort groupings would increase the number of students whose questions may be included in an exam. This could greatly increase the amount of viable exam data provided by this adjacency matrix approach.

7 Conclusion

Human performance or judgment data are indispensable for the evaluation of NLP systems. However, it is excruciatingly expensive to create such data sets from scratch [Jurafsky and Martin, 2008]. It is, therefore, advantageous to explore all possibilities of reuse. I present an elegant, matrix-based method for data reconfiguration that could be used on arbitrary human judgment, performance data or stimulus materials. An application of this method is demonstrated in the area of Question Generation.

I build exams out of sets of questions that have been answered by some students. An *exam* is a set of questions answered by the same students, so the goal is to seek the students who have answered the most questions in common.

Creating exam questions is a time-consuming process that is dependent on a feedback loop of how students perform on the exams. Finding perfect training data is also difficult and motivates reuse wherever possible. I have presented one method to create valuable data out of larger question sets using adjacency matrices. These exam matrices recreate exam-like data from sets of questions subsets of students have answered.

This approach could be extended to other pre-and post-processing of data sets, particularly those that are based on human judgments.

Acknowledgements

I would like to thank Prof. Bonnie Webber, A. Raymond Milowski, Dr. Simone Teufel, Dr. Donald MacDonald, Dr. Paul Denny, Dr. Joe Michael Kniss, Raymond Yuen and Jonathan Millin for their assistance with this work.

References

Aho, A., Hopcroft, J. and Ullman, J. 1974. *Design and Analysis of Computer Algorithms.* Addison-Wesley.

Bondy, J. A. and Murty, U. S. R. 1976. *Graph Theory with Applications*. North Holland.

Denny, Paul. 2009. PeerWise. http://peerwise.cs.auckland.ac.nz/.

Gronlund, Norman E. 1981. *Measurement and Evaluation in Teaching. 4th ed.*, Macmillan.

Jurafsky, D. and Martin, J. H. 2008. *Speech and Language Processing, 2nd ed.* Prentice Hall.

Mitkov, R.; Ha, L. A.; Varga, A.; and Rello, L. 2009. Semantic Similarity of Distractors in Multiple-Choice Tests: Extrinsic Evaluation. *In Proceedings of the EACL 2009 Workshop on GEometrical Models of Natural Language Semantics (GEMS)*, Athens, Greece, March 2009: 49-56.