Extracting Causal Relations in Deep Knowledge Tracing

Kevin Hong
University of California, Los Angeles
kevinhong1167@ucla.edu

Kia Karbasi
University of California, Los Angeles
kiakarbasi@ucla.edu

Gregory Pottie
University of California, Los Angeles
pottie@ee.ucla.edu

ABSTRACT

A longstanding goal in computational educational research is to develop explainable knowledge tracing (KT) models. Deep Knowledge Tracing (DKT), which leverages a Recurrent Neural Network (RNN) to predict student knowledge and performance on exercises, has been proposed as a major advancement over traditional KT methods. Several studies suggest that its performance gains stem from its ability to model bidirectional relationships between different knowledge components (KCs) within a course, enabling the inference of a student's understanding of one KC from their performance on others. In this paper, we challenge this prevailing explanation and demonstrate that DKT's strength lies in its implicit ability to model prerequisite relationships as a causal structure, rather than bidirectional relationships. By pruning exercise relation graphs into Directed Acyclic Graphs (DAGs) and training DKT on causal subsets of the Assistments dataset, we show that DKT's predictive capabilities align strongly with these causal structures. Furthermore, we propose an alternative method for extracting exercise relation DAGs using DKT's learned representations and provide empirical evidence supporting our claim. Our findings suggest that DKT's effectiveness is largely driven by its capacity to approximate causal dependencies between KCs rather than simple relational mappings.

Keywords

Knowledge Tracing, Causal Dependencies, Exercise Relations, Educational AI

1. INTRODUCTION

Computer-assisted educational technology, such as intelligent tutoring systems [9, 11], enables personalizing activities to suit individuals with varying levels of proficiency. A key component of these adaptive systems is knowledge tracing (KT), which aims to estimate students' mastery of several exercise concepts, known as knowledge components (KCs), as they interact with the corresponding exercises [2, 5]. For-

mally, $x_i = \{e_t, a_t\}$ represents a student's answer pair, where e_t represents the exercise ID, and a_t represents whether the student answered correctly or incorrectly. Given a series of past interactions, $X = \{x_1, x_2, \ldots, x_t\}$ and the next concept exercise, e_{t+1}, the task of the KT model is to estimate the likelihood of the student answering correctly, a_{t+1} [6]. Traditionally, Bayesian Knowledge Tracing (BKT) [6] was used to perform KT. Despite its popularity, BKT is often critiqued for its binary representation of knowledge states (mastered or not mastered) which oversimplifies the learning process. Several models were later proposed to address these limitations. Learning Factors Analysis [4] improved upon BKT by representing learning as a continuous process influenced by multiple exercise interactions and accumulated practice. Performance Factors Analysis [10] further captured the complexity of learning by tracking the effects of correct and incorrect prior exercise attempts on performance. These developments laid the groundwork for Deep Knowledge Tracing (DKT) [15], which popularized a key extension to KT: the ability to implicitly infer relationships between exercises.

The relationships between exercises can take the form of prerequisite dependencies, where understanding one exercise improves performance on another, or corequisites, where exercises depend on each other. Researchers have demonstrated that exercise relationships can be mapped into an exercise graph by analyzing student learning data. An exercise graph represents the dependencies between different concept exercises based on student learning patterns. Creating accurate exercise graphs allows educators to better sequence lessons and ensure that students master foundational concepts before progressing to more advanced ones. KT models are preferred for this task because they learn temporal patterns in student data to capture how mastery of one exercise affects success on another [5, 15, 17]. This enables KT models to uncover learning dependencies that content-based or expert-defined methods may overlook. Furthermore, automating the discovery of exercise relations through the use of KT models offers a scalable alternative to manually defining these relationships.

The work of DKT made a significant contribution to exercise relation discovery [15]. Their method assigns an influence score J_{ij} to every directed pair of exercises i and j, which is the conditional probability of correctly answering exercise j after correctly answering exercise i in the previous timestep, normalized by the sum of such condi-

tional probabilities. The influence score from i to j quantifies the prerequisite dependency of the concept i on the learning concept j. We refer to this method as the DKT method. DKT's superior performance can be attributed to the model's ability to learn these influence-based dependencies to estimate student knowledge. Many studies have incorporated exercise relations into subsequent KT models to improve performance and enhance interpretability in the prediction process. For example, Hierarchical Graph Knowledge Tracing (HGKT) [17], Structure-based Knowledge Tracing (SKT) [18], and Deep Knowledge Tracing with Multiple Relations (DKTMR) [7] are models that track both prerequisite and corequisite relationships between exercises. Graph Attention Knowledge Tracing (GAKT) [19] uses a graph attention network to uncover the underlying structure between exercises, learning these relationships using the method introduced in DKT. Graph-based Knowledge Tracing (GKT) [14] explores multiple statistics-based and learning-based methods to infer the latent graph structure and use the learned graph to perform KT. Among the statistics-based and learning-based methods evaluated, the DKT method to generate graphs performed the best when evaluating performance on the Assistments 2009 [8] dataset.

The original DKT work introduced influence scores to describe dependencies between exercises. In our work, we build on this idea by reinterpreting these dependencies as a causal structure. We formalize this by pruning exercise relations graphs into Directed Acyclic Graphs (DAGs) to reflect prerequisite relations, and show that DKT's predictive performance improves when trained on data filtered through these causal structures. We also introduce an alternative method for extracting exercise relations that yields accurate and more stable representations of student knowledge and underlying concept dependencies. To facilitate future research on these ideas, we have published our code [1].

2. METHODOLOGY

We conduct our study using the Assistments datasets, which are among the largest publicly available KT datasets and are widely used as benchmarks for KT models [1, 8, 16]. These datasets capture student interactions over extended periods of time and across a wide range of exercises in grade school mathematics. We utilize three datasets: Assistments 2009 [2] (*skill builder data corrected collapsed*), Assistments 2012 [3] (*2012-2013 data with predictions 4 final*), and Assistments 2017 [4] (*anonymized full release competition dataset*). The 2009 dataset contains 346,860 exercise attempts from 4,217 students across 123 exercises. The 2012 dataset includes 6,123,270 attempts from 46,674 students across 265 exercises. The 2017 dataset consists of 942,816 attempts from 1,709 students across 102 exercises. The datasets do not contain any personal information.

---

[1] https://github.com/kevinhongca/
dkt-causal-relations
[2] https://sites.google.com/site/assistmentsdata/
home/2009-2010-assistment-data/
skill-builder-data-2009-2010
[3] https://sites.google.com/site/assistmentsdata/
2012-13-school-data-with-affect
[4] https://sites.google.com/view/
assistmentsdatamining/dataset

We use DKT, implemented via the pyKT library [13], to learn the latent exercise relations and help generate the graphs. We begin by training a DKT model on each of the three Assistments datasets. Then, we switch the models to evaluation mode and apply the DKT method. To ensure the resulting graph represents a causal structure, we apply a minimum threshold to the influence scores. This filtering step removes weaker edges that could introduce cycles, allowing us to construct a DAG. Since the distribution of influence scores varies across the three datasets, we select the minimum dataset-specific thresholds that enforce acyclicity. We use a threshold of 0.0107 for Assistments 2009, 0.0051 for Assistments 2012, and 0.0139 for Assistments 2017.

Using the exercise relation DAG, we create a causal subset of the Assistments dataset by filtering interactions to include only those involving exercises with at least one incoming or outgoing edge. This ensures that the subset consists exclusively of exercises with learned causal relationships, which allows us to isolate and evaluate the influence of these causal structures. We then retrain the DKT model on this subset and evaluate its predictive accuracy. To isolate the impact of causal structure, we generate five random subsets of the dataset with bidirectional relations. Each random subset contains the same number of exercise concepts as its corresponding DAG-based subset, but the exercises are selected randomly rather than based on causal structure. The number of exercise concepts for the subsets was 60 for Assistments 2009, 83 for Assistments 2012, and 17 for Assistments 2017. We then train a separate DKT model on each random subset and compute the mean and standard deviation of their AUC scores. Finally, we compute a z-score to compare the AUC of the DAG-based subset against the distribution of AUC scores from the random subsets across the three Assistments datasets. This allows us to assess the influence of causal relationships on predictive performance.

Recognizing a potential limitation in DKT's standard relation extraction method, where influence scores are computed based on a single correct response per exercise, we propose a modified approach that simulates a student repeatedly answering the same type of exercise correctly until the model's estimated knowledge of that exercise stabilizes. Rather than relying on a one-time response, we feed multiple consecutive correct answers for a given exercise, allowing the model to iteratively update its estimate of the student's mastery of all the exercises.

Let $\hat{K}_t$ represent the student's estimated knowledge level of an exercise after the student has answered correctly t times. We continue feeding correct responses until the estimated mastery stabilizes according to the following criterion:

$$\hat{K}_t = \hat{K}_{t-1}, \quad \forall t \in [t_0 - T, t_0] \text{ s.t. } t_0 \geq T,$$

where if the model's estimate does not change for T consecutive iterations, we stop feeding additional responses and take $\hat{K}_t$ as the final knowledge estimate. In our experiments, we set $T = 100$, chosen heuristically as a reasonable value to allow the knowledge estimates to stabilize. Then, inspired by the method proposed by Piech et al., we use a modified

Table 1: AUC results for all Assistments subsets, divided into DAG-based causal subsets (Causal) and random subsets (Random). Causal subsets are derived using DKT's method.

Dataset	AUC	Exercises (#)	z-score
2009 Causal	0.905	60	3.50
2009 Random	0.859 ± 0.013	60	–
2012 Causal	0.727	83	1.50
2012 Random	0.721 ± 0.004	83	–
2017 Causal	0.718	17	1.02
2017 Random	0.704 ± 0.011	17	–

Table 2: AUC results for all Assistments subsets, divided into DAG-based modified causal subsets (MC) and random subsets (Random). Modified causal subsets are derived using Equation (1).

Dataset	AUC	Exercises (#)	z-score
2009 MC	0.875	68	1.67
2009 Random	0.851 ± 0.014	68	–
2012 MC	0.758	90	5.93
2012 Random	0.721 ± 0.004	90	–
2017 MC	0.712	59	1.85
2017 Random	0.703 ± 0.005	59	–

approach,

$$J_{ij} = \frac{z(j|i)}{\sum_k z(k|i)} \tag{1}$$

where $z(j|i)$ is the correctness probability assigned to exercise j on the next timestep given that the student answered exercise i correctly t_0 times on the previous t_0 timesteps. This iterative process provides a more stable estimate of concept relationships by reducing the influence of single-response variations. However, we acknowledge that this approach has practical limitations, which we discuss in Section 4.

We evaluate the effectiveness of our modified relation extraction method by repeating the methodology described earlier in this section using Equation 1 to generate newly constructed causal graphs: we extract the new DAG-causal subset, retrain DKT on this subset, and compare its AUC scores against those of randomly selected subsets. Note that when extracting the new DAG-causal subset, we again applied dataset-specific thresholds to the influence scores: 0.0129 for Assistments 2009, 0.0067 for Assistments 2012, and 0.0167 for Assistments 2017. By comparing results from both relation extraction methods, we assess whether probing more can yield more accurate causal relationships that help improve KT performance.

3. RESULTS

Across all three Assistments datasets, the causal subsets derived from the DKT method [15] consistently achieved higher AUC scores than the mean AUC of randomly selected subsets. See Table 1. This suggests that well-defined causal knowledge dependencies improve a KT model's ability to trace student learning, and that KT models appear to learn causal relationships more easily than bidirectional ones. Moreover, these findings introduce a potential evaluation metric for knowledge structure graphs: knowledge structures that more accurately capture prerequisite relationships may yield higher AUC scores when their corresponding concepts are used for training.

We then applied our new proposed method to find directed relationships, and as earlier, the new causal subsets consistently outperformed the mean AUC of the randomly selected subsets. See Table 2. To quantitatively compare the performance of the DKT method and our new method, we

compute the z-scores for all causal subsets, defined as:

$$z = \frac{\mathrm{AUC_{causal}} - \mu_{\mathrm{random}}}{\sigma_{\mathrm{random}}},$$

where $\mathrm{AUC_{causal}}$ represents the AUC of the causal subset, while μ_{random} and σ_{random} denote the mean and standard deviation of the AUCs obtained from the random subsets, respectively. We observe that the z-scores for our new method are higher than those of the original DKT method for two out of the three datasets. Furthermore, the average z-score is significantly higher for our proposed method. This suggests that our modified approach may yield more accurate representations of underlying knowledge structures.

To interpret our results further, we use Equation (1) to generate and analyze the directed exercise relation graphs learned by our models. Figure 1 presents the graph for Assistments 2009, where node numbers correspond to exercise IDs. To better identify the topics between various exercise groups, we apply the algorithms described in [3, 12] to modulate the exercises into topics and use color coding to visually distinguish them. Overall, our approach effectively reveals a meaningful causal structure. Focusing on graphical data topics, we observe that Exercise 2 (Pie Charts 1) serves as a prerequisite for multiple exercises. Since pie charts rely heavily on percentage calculations, it follows that Exercise 2 also contributes to learning related topics, such as Venn Diagrams represented as percentages (Exercise 70) and proportion calculations (Exercise 79). Additionally, the model successfully captures the progression of concepts toward more advanced topics. For example, Exercise 2 (Pie Charts 1) is a prerequisite for Exercise 37 (Pie Charts 2), reinforcing the importance of mastering basic pie chart concepts before progressing. Furthermore, both Exercises 2 and 37 serve as prerequisites for Exercise 48 (Pie Charts 3). While the figure we show provides valuable insights, not all directed edges necessarily indicate true prerequisite relationships, as some connections may result from indirect correlations rather than direct dependencies. Despite these occasional inaccuracies, our method still captures an overall meaningful structure.

4. DISCUSSION

In this section, we provide insights into the rationale behind our proposed method for extracting directed relations. We use a DKT model trained on the full Assistments 2009
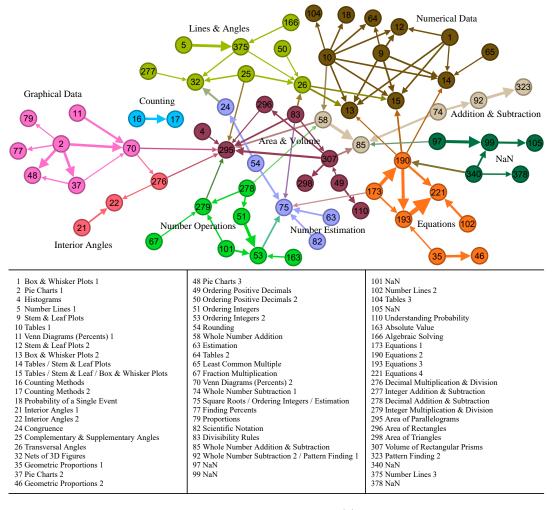
Lines & Angles
Numerical Data
Graphical Data
Counting
Addition & Subtraction
Area & Volume
NaN
Number Operations
Number Estimation
Equations
Interior Angles

1 Box & Whisker Plots 1	48 Pie Charts 3	101 NaN
2 Pie Charts 1	49 Ordering Positive Decimals	102 Number Lines 2
4 Histograms	50 Ordering Positive Decimals 2	104 Tables 3
5 Number Lines 1	51 Ordering Integers	105 NaN
9 Stem & Leaf Plots	53 Ordering Integers 2	110 Understanding Probability
10 Tables 1	54 Rounding	163 Absolute Value
11 Venn Diagrams (Percents) 1	58 Whole Number Addition	166 Algebraic Solving
12 Stem & Leaf Plots 2	63 Estimation	173 Equations 1
13 Box & Whisker Plots 2	64 Tables 2	190 Equations 2
14 Tables / Stem & Leaf Plots	65 Least Common Multiple	193 Equations 3
15 Tables / Stem & Leaf / Box & Whisker Plots	67 Fraction Multiplication	221 Equations 4
16 Counting Methods	70 Venn Diagrams (Percents) 2	276 Decimal Multiplication & Division
17 Counting Methods 2	74 Whole Number Subtraction 1	277 Integer Addition & Subtraction
18 Probability of a Single Event	75 Square Roots / Ordering Integers / Estimation	278 Decimal Addition & Subtraction
21 Interior Angles 1	77 Finding Percents	279 Integer Multiplication & Division
22 Interior Angles 2	79 Proportions	295 Area of Parallelograms
24 Congruence	82 Scientific Notation	296 Area of Rectangles
25 Complementary & Supplementary Angles	83 Divisibility Rules	298 Area of Triangles
26 Transversal Angles	85 Whole Number Addition & Subtraction	307 Volume of Rectangular Prisms
32 Nets of 3D Figures	92 Whole Number Subtraction 2 / Pattern Finding 1	323 Pattern Finding 2
35 Geometric Proportions 1	97 NaN	340 NaN
37 Pie Charts 2	99 NaN	375 Number Lines 3
46 Geometric Proportions 2		378 NaN

Figure 1: Assistments 2009 DAG graph of exercise relations using Equation (1)**. Arrow weight indicates prerequisite connection strength. Topic labels are manually added and color coded.**

dataset and set it to evaluation mode to predict the knowledge state of a new simulated student. Our goal is to identify a student's top three highest-ranked exercise concepts, known as KCs, sorted by estimated knowledge mastery, after correctly completing concept 278 (Decimal Addition & Subtraction). Using the original method, we determine the student's top three highest predicted KCs immediately after answering concept 278 correctly. In our approach, we allow the student's understanding of concept 278 to stabilize through one hundred correct responses before extracting the top three highest concept estimates. In other words, we seek three values of j that maximize $y(j|i)$ and three that maximize $z(j|i)$, given that $i = 278$ and $i \neq j$.

As shown in Table 3, the top three predicted KC masteries for concept 278 differ between the DKT method and our new approach. The highest-ranked concepts from the DKT method are primarily geometry-related, whereas our approach identifies exercises that are more directly relevant to concept 278, all involving integer operations. We observe that it only takes three consecutive correct responses of exercise 278 for the top three orderings to stabilize. Because $y(j|i)$ and $z(j|i)$ correspond to the numerators of the DKT

method and our method respectively, these values directly affect how exercise relations are constructed. Referring back to Figure 1, we see that our method indeed helps relate exercise 278 to the three exercises shown in the Table.

While our modified method may provide a better alternative to the method proposed by DKT, a key limitation of our method is that, in practice, students typically do not answer more than a few questions per exercise. The assumption that a student can be prompted up to 100 times on a given exercise is unlikely to reflect real-world learning scenarios. Future work should experiment with the amount of probing using a more practical number of responses (e.g. 5) or to employ a convergence-based stopping criteria to terminate the iterations when the difference between successive knowledge estimates falls below a certain threshold.

Another limitation concerns the evaluation setup used to compare DAG-based causal subsets and random subsets. The causal subsets and random subsets may differ in ways beyond their corresponding graph structure. Although we ensure that both subsets contain the same number of exercises, they may still vary in concept coverage, exercise diffi-

Table 3: Comparison between the top three KC masteries between DKT's method and our approach

DKT Method			Modified Approach		
Exercise ID	Exercise Name	Mastery	Exercise ID	Exercise Name	Mastery
24	Congruence	0.677	51	Ordering Integers	0.842
307	Volume of Rectangular Prisms	0.673	279	Integer Multiplication & Division	0.839
26	Transversal Angles	0.667	58	Whole Number Addition	0.815

culty, or the types of students engaging with the exercises. As a result, some portion of the observed performance gains may be due to differences in exercise or student characteristics, rather than structure alone. Future work should explore more controlled subset construction, such as concept-matched sampling, in which random subsets reflect the distribution of concept types found in the DAG-based subset. For example, if the DAG-based subset consists of 40% arithmetic, 30% geometry, and 30% probability exercises, the corresponding random subset could try to maintain a similar distribution. Strategies like this will help better isolate the impact of causal structure on model performance.

5. CONCLUSION

We show that DKT achieves better predictive performance when trained on DAG exercise subsets, suggesting it effectively learns causal concept dependencies. We introduce a novel method for calculating influence scores that stabilizes knowledge estimates and helps construct accurate exercise relation graphs. Finally, we acknowledge the limitations regarding the practicality of repeated probing and the need for more controlled experimental comparisons.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] G. Abdelrahman, Q. Wang, and B. Nunes. Knowledge Tracing: A Survey. *ACM Comput. Surv.*, 55(11):224:1–224:37, Feb. 2023.

[2] Y. Bai, J. Zhao, T. Wei, Q. Cai, and L. He. A Survey of Explainable Knowledge Tracing, Mar. 2024. arXiv:2403.07279 [cs].

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct. 2008. arXiv:0803.0476 [physics].

[4] H. Cen, K. Koedinger, and B. Junker. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, pages 164–175, Berlin, Heidelberg, 2006. Springer.

[5] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian. Prerequisite-Driven Deep Knowledge Tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48, Singapore, Nov. 2018. IEEE.

[6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec. 1994.

[7] Z. Duan, X. Dong, H. Gu, X. Wu, Z. Li, and D. Zhou. Towards more accurate and interpretable model: Fusing multiple knowledge relations into deep knowledge tracing. *Expert Systems with Applications*, 243:122573, June 2024.

[8] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19(3):243–266, Aug. 2009.

[9] S. Feng, A. J. Magana, and D. Kao. A Systematic Review of Literature on the Effectiveness of Intelligent Tutoring Systems in STEM. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Lincoln, NE, USA, Oct. 2021. IEEE.

[10] P. I. P. Jr, H. Cen, and K. R. Koedinger. Performance Factors Analysis – A New Alternative to Knowledge Tracing.

[11] J. A. Kulik and J. D. Fletcher. Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review. *Review of Educational Research*, 86(1):42–78, Mar. 2016. Publisher: American Educational Research Association.

[12] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian Dynamics and Multiscale Modular Structure in Networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90, July 2014. arXiv:0812.1770 [physics].

[13] Z. Liu, Q. Liu, J. Chen, S. Huang, J. Tang, and W. Luo. pyKT: A Python Library to Benchmark Deep Learning based Knowledge Tracing Models, Jan. 2023. arXiv:2206.11460 [cs].

[14] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 156–163, Thessaloniki Greece, Oct. 2019. ACM.

[15] C. Piech, J. Spencer, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep Knowledge Tracing, June 2015. arXiv:1506.05908 [cs].

[16] E. Prihar, Manaal Syed, Korinn Ostrow, S. Shaw, A. Sales, and N. Heffernan. Exploring Common Trends in Online Educational Experiments. July 2022. Publisher: Zenodo.

[17] H. Tong, Z. Wang, Y. Zhou, S. Tong, W. Han, and Q. Liu. HGKT: Introducing Hierarchical Exercise Graph for Knowledge Tracing, Aug. 2022. arXiv:2006.16915 [cs].

[18] S. Tong, Q. Liu, W. Huang, Z. Hunag, E. Chen,

C. Liu, H. Ma, and S. Wang. Structure-Based Knowledge Tracing: An Influence Propagation View. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 541–550, Nov. 2020. ISSN: 2374-8486.

[19] Z. Zhao, Z. Liu, B. Wang, L. Ouyang, C. Wang, and Y. Ouyang. Research on Deep Knowledge Tracing Model Integrating Graph Attention Network. In *2022 Prognostics and Health Management Conference (PHM-2022 London)*, pages 389–394, May 2022. ISSN: 2166-5656.