

# DEVELOPMENT OF LARGE SCALE STEM PROBLEM DATABASES FOR STUDENT LEARNING AND ASSESSMENT TOOLS

J. A. Davis and S. A. Lorimer  
Grant MacEwan University  
davisj80@macewan.ca

**Abstract** – Problem databases in STEM courses are used in tools for the development of student learning and final assessment. In addition, large problem databases are used to develop models for automatic assessment and feedback of students' work. However, the availability of large, open source, problem databases for specific courses is limited, and in-house development of a wide variety of problems can take years. In this paper, the framework for a problem database in STEM courses was created using semantic analysis of sentence structure and composition. Problem statements were analyzed to determine the key grammatical constructs that are used in commonly posed problems. Based on this analysis, software was developed to create large problem databases which allow for simple extension to other courses. Using a first-year mechanics course this software was populated with a few generalized question and sentence structures to create a large problem database.

**Keywords:** Large Problem Databases, Problem Statements

## 1 INTRODUCTION

Effective student learning and assessment tools are useful for STEM courses and are thought to help students learn and improve student self-efficacy. One method that students become more competent with material, in first year STEM courses, is by practicing learned concepts through assignments [1] and practice exams [2]. Re-use of assignments can begin to be ineffective when solutions become available and easily found (or solved by others) on the internet and when students use these solutions to passively learn concepts as opposed to actively solving problems. Other tools used in computer adaptive teaching are useful for learning, allowing students to practice questions at individualized difficulty levels [3]. A common feature in these assessments/learning tools is the database of problems from which the learning item is made.

There has been a growing interest in the development and use of Machine learning models in the classroom setting (see [4-6] for example). Models developed using databases, which include course concepts, are useful in a

classroom environment in a variety of ways including the development of tools for assignment and solution generation, automatic marking, and automatic feedback to students. To be effective, however, databases need to have a large number of training data which cover a diverse range of problem types. Some problem databases exist, such as Aqua-Rat [7] which contains (~100k) algebraic word problems or SQuAD [8] which consists of general questions (~150k). Successful online software systems for developing and managing assessments, such as WeBWork [9], exist and have been used to develop questions for engineering courses [10-12]. This software uses the problem database Open Problem Library which includes ~35k problem templates over a variety of courses [13]. One constraint with Open Problem Library is that it is not currently straightforward to create a large independent problem database from the library containing the problem templates.

The goals of this paper are to develop a framework for the creation of large scale problem databases for STEM problems which can be then used in learning and assessment tools and to populate and create a large problem database for a first-year Engineering Mechanics course.

## 2 METHODS

In general, a problem database contains a diverse range of problem statements. A problem statement is defined here as multiple sentences in which there exists at least one problem. Similarly, a problem is defined here as a sentence (or part of a sentence) which is in either imperative or interrogative form and implicitly contains an action verb, an unknown, and potentially knowns and constraints. The current study is restricted to problems without accompanying figures or tables.

The first step in the development of a problem database is to define the equations which link the physical and geometric variables for the topic being considered. Table 1 shows a subset of the equations (restricted to equations related to force and moment vectors) linking physical equations for forces, moments, position, and unit vectors used in a first year mechanics course.

A significant number of the equations are general and can be used on various types of vectors (eg. vector magnitude from a force or moment). Here, the equations may also contain geometric variables (distances and angles). Equations for geometric variables, however, are problem specific and depend on the geometry of the objects involved.

**Table 1:** Equations for vectors

Type	Equations (ID)
General	$ \chi  = \sqrt{\chi_x^2 + \chi_y^2 + \chi_z^2}$ (0), $\bar{\chi} =  \chi \bar{u}$ (1), where $\chi \in \{F, r, M, u\}$
Unit	$\bar{u} = \cos(\theta)\bar{i} + \sin(\theta)\bar{j}$ (2), $\bar{u} = \cos(\alpha)\bar{i} + \cos(\beta)\bar{j} + \cos(\gamma)\bar{k}$ (3) $\cos(\alpha)^2 + \cos(\beta)^2 + \cos(\gamma)^2 = 1$ (4)
Position	$\vec{r}_{ab} = (P_{x,b} - P_{x,a})\bar{i} + (P_{y,b} - P_{y,a})\bar{j}$ (9)
Moment	$\vec{M} = \vec{r} \times \vec{F}$ (A)

A sample geometry can be found in Table 2 along with a subset of its geometric relations. Typically, geometric equations are based on supplementary and complementary angles, interior angles in polygons, trigonometric relations, Sine law, and Cosine law. In addition, Tables 1 and 2 provide a single Hexadecimal character which can be used to identify the equation. From the given equations, in Tables 1 and 2, the next step in the development of a problem database is to determine the

**Table 2:** Equations resulting for a specific geometry

Geometric Relations (ID)
$\theta + \phi = 90$ (5), $\theta + \psi + \xi = 180$ (6)
$r_{14}^2 = r_{12}^2 + r_{24}^2 - 2r_{12}r_{24}\cos(\xi)$ (7),
$r_{14}\sin(\xi) = r_{24}\sin(\theta)$ or $r_{12}\sin(\psi) = r_{14}\sin(\xi)$ (8)

different combinations of unknowns and knowns in the form  $\vec{F} \leftarrow |F|, \bar{u}$  where  $\vec{F}$  represents the unknown and  $|F|, \bar{u}$  represent the problem knowns. Physically, this equation represents the problem of solving for a force vector given its magnitude and direction. Table 3 shows a non-exhaustive list of the variables developed from the physical and geometric equations. Here an identification code is also given which provides information on the number of equations required to solve the unknowns (number of symbols) and the different equations needed to solve the problem. As an example  $\vec{F} \leftarrow |F|, \theta$  would be given the ID 12 as it would require equations 1 and 2 (as defined in Table 1) to solve for  $\vec{F}$ . This may also be useful to give an indication of the problem's difficulty level (based on the number and type of equations used to

**Table 3:** List of a subset of the known combinations for various unknown vector quantities

$\vec{F}$		$\vec{r}$		$\bar{u}$		$\vec{M}$	
Known	ID	Known	ID	Known	ID	Known	ID
$ F , \bar{u}$	1	$ r , \bar{u}$	1	$\theta$	2	$ M , \bar{u}$	1
$ F , \theta$	12	$P_A, P_B$	9	$\phi$	25	$\vec{r}, \vec{F}$	A
$ F , \phi$	125	$ r , \theta$	12	$\vec{F}$	01	$ M , \theta$	12
$ F , \vec{r}$	011	$ r , \phi$	125	$\vec{M}$	01	$\vec{F},  r , \bar{u}$	1A
$ F , \alpha, \beta$	134	$ r , \vec{F}$	011	$\vec{r}$	01	$ M , \alpha, \beta$	134
$ F , \alpha, \gamma$	134	$ r , \alpha, \beta$	134	$\alpha, \beta \mid \alpha, \gamma$ $\mid \beta, \gamma$	34	$ M , \alpha, \gamma$	134
$ F , \beta, \gamma$	134	$ r , \alpha, \gamma$	134	$\psi, \xi$	26	$ M , \beta, \gamma$	134
$ F , \psi, \xi$	126	$ r , \beta, \gamma$	134	$\vec{r}, \vec{F}$	01A	$\vec{F},  r , \theta$	12A
		$ r_{14} , \psi, \xi$	126	$P_A, P_B$	019		
		$ r_{14} ,  r_{24} , \xi$	1268	$ r_{12} ,  r_{24} , \psi$	268		

solve the problem). It should be noted that this setup ( $\vec{F} \leftarrow |F|, \bar{u}$ ) can be used to create problem statements with a variety of different problem solution methodologies (ex. plug and chug vs. guess and check). The only constraint is that there must be at least one known and exactly one unknown. After determining the different combinations of problems in mathematical form, the next step involves putting the problems into sentence form. Table 4 shows examples with different levels of increasingly complex sentence structures for the problem  $\vec{F} \leftarrow |F|, \bar{u}$  with the corresponding structure broken into the respective grammatical tags. The simplest sentences are formed by adding a problem action verb (VERB) and, in certain sentences, a subordinating conjunction (SCONJ). Problem action verbs are verbs used in

**Table 4:** Example sentence structure

Level	Structure	Example
1	[SCONJ] [K <sub>1</sub> ] and [K <sub>2</sub> ], [VERB] [U]. [VERB] [U] [SCONJ] [K <sub>1</sub> ] and [K <sub>2</sub> ]. [VERB] [U]. [IVRB] [K <sub>1</sub> ] and [K <sub>2</sub> ].	Given $ F =5$ kN and $\vec{u}=1\vec{i}+0\vec{j}$ , determine $\vec{F}$ . Determine $\vec{F}$ given $ F =5$ kN and $\vec{u}=1\vec{i}+0\vec{j}$ . Determine $\vec{F}$ . Use $ F =5$ kN and $\vec{u}=1\vec{i}+0\vec{j}$ .
2	[SCONJ] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ], [VERB] [PROPN][U]. [VERB] [PROPN][U] [SCONJ] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ]. [VERB] [PROPN][U]. [IVRB] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ].	Given the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ , determine the force vector $\vec{F}$ . Determine the force vector $\vec{F}$ given the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ . Determine the force vector $\vec{F}$ . Take the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ .
3	[SCONJ] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ], [VERB] [PROPN][U] [PP]. [VERB] [PROPN][U] [PP] [SCONJ] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ]. [VERB] [PROPN][U][PP]. [IVRB] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ]. [PROPN][PP]. [SCONJ] [PROPN <sub>1</sub> ][K <sub>1</sub> ] and [PROPN <sub>2</sub> ][K <sub>2</sub> ], [VERB] [PROPN][U].	Given the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ , determine the force vector $\vec{F}$ acting on the block. Determine the force vector $\vec{F}$ acting on the block given the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ . Determine the force vector $\vec{F}$ acting on the block. Take the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ . A force acts on the block. Given the magnitude of the force $ F =5$ kN and the unit vector $\vec{u}=1\vec{i}+0\vec{j}$ , determine the force vector $\vec{F}$ .

imperative sentences which are based on Bloom’s taxonomy [14] where different groups of verbs are used based on the cognitive requirements of the task. For a first-year STEM course “application” level action verbs are typically found and include verbs such as: calculate, compute, determine, and solve. For each problem statement, the knowns (K) are composed of a variable, numeric values, and units (eg.  $|F|=5.3$  [N]). Each known would have a range of numeric values for the specific variable in question which would be dependent on the units used (or the scale of the application in the problem). The unknowns (U) consist of a variable and a unit (eg.  $\vec{F}$  [N]). At the first level, Table 4 shows the initial structure of the sentences created for the database. At this stage the constructs are mathematical and do not provide much insight into the context of the problem being solved. In applied fields such as engineering or as an attempt to engage students more, problems are typically based on real-world applications. Proper nouns (PROPN), in level 2, and prepositional phrases (PP), in level 3, are then

added to the sentence structure to bring in more meaning to the problems. The additional verbs, subordinate conjunctions, proper nouns, and prepositional phrases also introduce variability to the sentences which depend on the unknowns and knowns used in the problem. This variability is created through sentence structure which depends on the location of the grammatical token in the sentence (ie. acts vs. acting) or through variation of the token in the language (ie. Moment vs. Torque). Table 5 lists the different grammatical tokens used for each of the variables. Here, variables are shown to have only a few proper nouns which describe them. An exception to this is when the variable can be sub-categorized. An example is Force where it can be subcategorized into frictional, gravitational, spring force (to name a few). Units can also have a large number of variations which depend on the application of the problem (which then sets the scale) or the dimensional system used (SI vs FPS). For a first year course, units are typically restricted to only a few variations.

**Table 5:** List of different sentence tokens used for Variables

Variable	Proper Noun [PROPN]	Prepositional Phrase [PP]	Units		Verb [VERB]	Subordinate Conjunction [SCONJ]
			SI	FPS		
$\vec{F}$	Force, Force vector, Applied force	acts at a point, acts on an [OBJ]	N, kN	lb, kip	Calculate, Compute, Determine, Solve for	Given, Provided, If, Taking
$ F $	Magnitude of the force, Force	acts at a point, acts on an [OBJ]				
$\vec{r}$	Position vector	acts between two points	mm, cm, m	in, ft		
$ r $	Magnitude of the position vector, Distance, Length	acts between two points				
$P_A$	Point, Coordinate Point	acts at point A				
$\vec{M}$	Moment, Moment vector, Torque	acts about a point	N.m, kN.m	lb.ft, kip.in		
$ M $	Magnitude of the moment, Moment, Torque	acts about a point				
$\vec{u}$	Unit vector, Direction vector	acts between two points	--			
$\alpha, \beta, \gamma$	Direction angle	measured between the vector and the axes	°, rad			
$\theta, \phi$	Angle	measured between [two lines, two surfaces, the surface and the vector, the line and the vector]				

Similarly, prepositional phrases can include great variation when including objects which are context specific to a given type of problem. An example is a force which can be applied to a wide variety of objects. Table 6 shows a small subset of the objects to which forces can be applied. When creating problem databases, in addition to creating problem statements, it is also important to

**Table 6:** Various Objects used in a Mechanics Course

Concept	Object [OBJ]
Force on a Body	Bars, Frames, Rods, Member, Plate, Bracket, Crate, Beam, Crane, Pipe, Gear, Pole
Force along a line	Rope, Rod, Chain, Wire, Cable

include training labels. These labels are useful for supervised learning algorithms in machine learning. Given a problem statement a trained machine learning model would be able to make a prediction. The labels provide the answer to that prediction. In terms of problem solving, some sample training labels could include: the subject (or topic) of the problem, knowns, unknowns, or the equations needed to solve the problem (ie. problem ID). More advanced labels could include labels useful in training Natural Language Processing libraries to provide

better prediction accuracy for sentences containing problems with knowns in them. Here, the knowns, unknowns, and problem ID are provided directly as labels in the databases. Indirectly, the labels: problems, and course subject area (Engineering Mechanics) could be added afterwards as the current database is restricted to only these types.

To create the database, Python programming language was used due to the ease of programming and the large availability of libraries. An overview of the program, using a UML class diagram, is shown in Fig. 1. Here, the main class is the ProblemGenerator class which first reads in the input files. To keep the program as general as possible, different input files were used to store information such as: the problem and equation codes (Table 3), the sentence structure (Table 4), unit information, variable ranges, and each variable's part of speech information (Tables 5 and 6). Once the files were read in, the information was restructured and stored in the appropriate classes: SentenceSchema (sentence structure), ProblemSchema (problem structure), UnitSchema (unit information), VariableSchema (numeric ranges of variables), and VariablePOSSchema (sentence structure combinations). The ProblemGenerator class then sets up all variable combinations and formats them for printing

which includes setting up variables using Unicode characters.

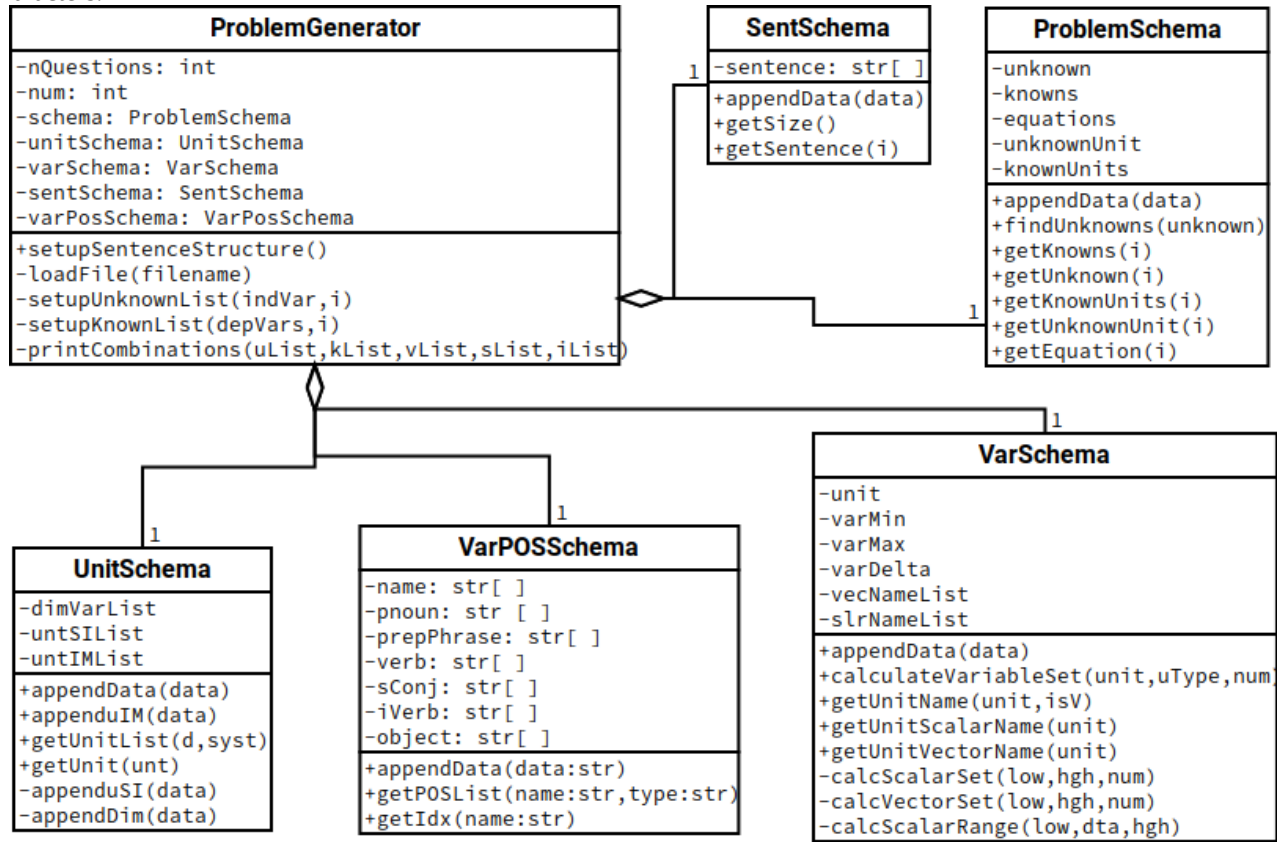


Fig. 1: UML Class Diagram for the Code

### 3 RESULTS

Once coded, the program was run to create a problem database with over 300 million problem statements using the 38 different problem types in Table 3 which resulted in 58 Gbytes of uncompressed data. Compressed, however, the data was reduced to 0.97 Gbytes in XZ compression format at a compression level of 6. For the problem database, variations in numerical values were restricted to only 3 variations for the knowns for all sentence structures of a given unit. A sample of the problem statements generated is shown in Table 7 with the Machine Learning labels removed. As an example calculation, for the problem  $|F| \leftarrow \vec{F}, \hat{u}$  with the sentence structure “[SCONJ][KList],[VERB][U]” an estimate of the number of combinations is  $[4][(3*1*3*4)*(2*1*3*1)] [4][3*1*4*12] = 497,664$  different combinations for the one sentence structure. This uses [U]=[PROPN][VAR][UNIT][PP] and [KList]=[K<sub>1</sub>] and [K<sub>2</sub>] where [K<sub>1</sub>]=[PROPN][VAR]=[NUM][UNIT]. With multiple sentence structures (at level 3 there are 4 sentence variations) the number of combinations grows with the number of knowns and the number of objects in the

prepositional phrases. For the first problem the number of total problem combinations when generated was

**Table 7:** Samples from the generated Problem Database

Taking $ F =586$ [N] and $\vec{u}=0.157i + 0.549j + 0.820k$ [ ], calculate $F$ [N].
Determine the force $F$ [N] acting on a gear. Take the force $ F =188$ [N], the direction angle $\alpha=100$ [°], and the direction angle $\beta=223$ [°].
Compute the position vector $\vec{r}_{12}$ [in] provided $P_1=[-5.01, 19.9, 6.37]$ [ft] and $P_2=[15.0, -9.59, -17.8]$ [ft].
Given the magnitude of the position vector $ r =408$ [in] and the force vector $\vec{F}=-0.205i + 1.48j + 0.783k$ [kip], determine $\vec{r}$ [in] acting between two points.
Provided $\theta=101$ [°], solve for the unit vector $\vec{u}$ [ ] .
Determine the unit vector $\vec{u}$ [ ] provided the moment $\vec{M}=2291i - 620j - 3268k$ [lb.ft].
A moment acts about a point. Provided the magnitude of the Torque $ M =423$ [kip.in] and $\vec{u}=0.664i + 0.685j + 0.297k$ [ ], determine the torque $\vec{M}$ [kip.in].

Taking  $\vec{F} = -194i + 479j - 288k$  [N],  $|r| = 551$  [cm], and the unit vector  $\vec{u} = 0.480i - 0.583j + 0.654k$  [ ], compute  $\vec{M}$  [N.m].

$n = 2,135,808$ . One reason for the large number of combinations was the diverse range of objects on which a force can act (which was restricted in this study to 12 different objects). Finally, Table 8 shows a sample of from the database which shows some of the problem variability for the sample problem  $\vec{F} \leftarrow |F|, \theta$ .

**Table 8:** Samples problem variability

Given $ F  = 141$ [N] and $\theta = 337$ [°], solve for $\vec{F}$ [N] acting on a beam.
Calculate $\vec{F}$ [N] acting on a member provided $ F  = 4.09$ [kN] and the angle $\theta = 5.59$ [rad].
A force acts on a pipe. If the force $ F  = 54.4$ [lb] and $\theta = 289$ [°], determine the force vector $\vec{F}$ [kip].
Solve for the force vector $\vec{F}$ [N] acting on a gear if $ F  = 54.4$ [lb] and the angle $\theta = 289$ [°].
A force acts on a frame. Given the magnitude of the force $ F  = 0.503$ [kip] and $\theta = 0.944$ [rad], compute the force vector $\vec{F}$ [kip].

#### 4 DISCUSSION AND CONCLUSIONS

The development of a framework for a large-scale problem database of STEM subjects has been summarized in this paper starting with the mathematical problem, then adding complexity and application through variations in sentence structure. At this point, structural elements in each sentence lend themselves to natural variations in proper nouns, prepositional phrases, action verbs, and subordinate conjunctions. This results in a potentially large number of total combinations of unique problem statements. For a particular teaching style an instructor may choose to limit the number of combinations by limiting the problems to certain categories of sets of units (ex. non-varying units in problems). However, if developing a machine learning model for student assessment is important then it may be necessary to capture all potential variations in problem statements in order to create a robust model.

By looking at development of a problem statement one can also gain insight into problem solving by looking at the inverse problem. Starting out with a complex sentence, allows one to extract information to produce the mathematical problem (ex.  $\vec{F} \leftarrow |F|, \vec{u}$ ). Helping students search for and extract essential information such as problem knowns, unknowns, and constraints as well as understanding the natural variations in posing of problems may help them learn and search for solution strategies more effectively. Furthermore, problem solving

methodology could potentially be integrated into first-year English courses when learning grammar by showing students the importance of different structural elements within problem statements.

There are also several applications where the large problem database presented in this study would be useful in traditional learning environments as well as for the development of machine learning models for assessment. The database can be used within tools to aid quiz, assignment, or exam question generation. The database can be used within tools to choose problems of similar type or difficulty levels, to focus on improving unit conversion or vocabulary, or to ensure that certain equations are used to test knowledge of learning outcomes. In addition, with respect to machine learning, the database can be used to develop models to help in the automatic marking and feedback of assessments through problem and solution methodology classification. For students, the database could be used within software to practice problems as an aid to help reinforce learning or to scaffold learning. Although, open source problem libraries already exist, the current database aids by providing a large database of problems which is stored as ready-made problem statements (including various tags) for use in teaching software and for the development of machine learning models. The resulting database offers an increase in accessibility of open source problems independent of any application or software. It also offers an increased inclusion in that the large database improves the existing sample sets of problems available which helps offset machine learning biases.

As an improvement and future work the database will be expanded to include solutions (and intermediate steps) to the problems. In addition, constraints to problems will be added as an option to the software to create more focused problem statements (ie. forces along a line). Next, the software will be tested with more problem types. With a problem database now developed, machine learning models can now be trained to: validate problem statements, help classify the problems and determine their solution methodology, and provide helpful hints. Finally, another improvement for the current work would be to create a graphical user interface to ease problem creation.

#### References

1. P. Wankat, "The Role of Homework," *Paper presented at the annual meeting for American Society for Engineering Education*, (Albuquerque, USA; June 24-27 2001), 2001.
2. K. Rawson and J. Dunlowsky, "When is Practice Testing Most Effective for Improving the Durability and Efficiency of Student Learning," *Educ. Psychol Rev*, vol. 24, pp 419-435, 2012.

- 3 M. Lilley, T. Barker, and C. Britton, "The development and evaluation of a software prototype for computer adaptive testing," *Computers and Education*, vol. 43, pp109-123, 2004.
- 4 A. Yahya, A. Osman, A. Taleb, A. Alattab, "Analyzing the Cognitive level of Classroom Questions Using Machine Learning Techniques," *Procedia – Social Behavioral Sciences*, vol. 97, no. 6, pp 587-595, 2013.
- 5 P. Goldberg, O. Sumer, K. Sturmer, W. Wagner, R. Gollner, P. Gerjets, E. Kasneci, U. Trautwein, "Attentive or not? Towards a Machine Learning Approach to Assessing Students' Visible Engagement in Classroom Instruction," *Educational Psychology Review*, vol. 33, pp 27-49, 2019.
- 6 S. Kotsiantis, C. Pierrakeas, P. Pintelas, "Predicting Students' performance in distance learning using machine learning techniques," *Applied Artificial Intelligence*, vol. 18, no. 5, pp 411-426, 2010.
- 7 W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program Induction by Rationale Generation : Learning to Solve and Explain Algebraic Word Problems," *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, vol. 1, pp. 158–167, 2017.
- 8 P. Rajpurkar, R. Jia, and P. Liang, 2018. "Know What You Don't Know: Unanswerable Questions for SQuAD." *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, vol. 2, pp. 784–789, 2018.
- 9 Mathematical Association of America. "WeBWorK" [Online]. Available: <https://webwork.maa.org/intro.html> [Accessed 25-April-2021].
- 10 J. Lee and J. Verrett, "WeBWorK as an Open Online Homework system in Material and Energy Balances," *Proc. of the 2018 Canadian Engineering Education Association Conference*, Paper 101, (Vancouver, Canada; June 3-6 2018), 2018.
- 11 M. Swanbom, D. Moller, and K. Evans, "Open-source, Online Homework for Statics and Mechanics of Materials Using WeBWorK: Assessing Effects on Student Learning," *Proc. Of the 123rd ASEE annual Conference*, Paper 16092, (New Orleans, USA; June 26-29, 2016), 2016.
- 12 A. d'Entremont, P. Walls, and P. Crompton, "Student Feedback and Problem Development for WeBWorK in a Second-Year Mechanical Engineering Program," *Proc. of the 2017 Canadian Engineering Education Association Conference*, Paper 033, Toronto, Canada; June 4-7 2017), 2017.
- 13 Mathematical Association of America. "Open Problem Library".[Online]. Available: [/wiki/Open\\_Problem\\_Library](https://webwork.maa.org/wiki/Open_Problem_Library) [Accessed 25-April-2021].
- 14 B. Bloom, M. Engelhart, E. Furst, W. Hill, and D. Krathwohl, *Taxonomy of educational objectives: The classification of educational goals. Handbok I: Cognitive domain*, New York, NY: David McKay Company, 1956.