“Majorly Adapted Translator”: Towards Major Adaptation in ITS

Tina Daaboul and Hicham Hage
Department of Computer Science, Notre Dame University - Louaize, Zouk Mosbeh, Keserwan, Lebanon

Keywords: Culturally Aware Learning System, Major Adaptation, Information Extraction, Intelligent Tutoring Systems, Relation Extraction.

Abstract: Culturally Aware Learning Systems are intelligent systems that adapt learning materials or techniques to the culture of learners having different “country, hobbies, experiences, etc.”, helping them better understand the topics being taught. In higher education, many learning sessions involve students of different majors. As observed, many instructors tend to manually modify the exercises several times, once for every major to adapt to the culture, which is tedious and impractical. Therefore, in this paper we propose an approach to making learning sessions adaptable to the major of the learner. Specifically, this work introduces an Artificial Intelligent system, “Majorly Adapted Translator (MAT)”, which aims at translating and adapting exercises from one major to another. MAT has two main phases, the first identifies the parts of an exercise that needs changing and creates an exercise template. The second translates and adapts the exercise. This work, highlights the first phase, the Feature Extract phase, which relies on our own relation extraction method to identify variables which extracts relations specific to named entities by using dependency relations and shallow parsing. Moreover, we report the performance of the system that was tested on a number of probability exercises.

1 INTRODUCTION

In recent years, culture became widely adapted especially on the level of educational technology, since an e-learning session can easily involve people from different countries and cultures. A shift was done in e-learning to become more culturally aware; specifically in intelligent tutoring systems (ITS), which are named “Culturally Adapted Tutoring Systems (CATS)”.

In the case of CATS, the culture being adapted is the “social culture” since it mainly involves factors on the social level such as country. However, learners taking the same topic might be from different majors, in this case adapting to the “social culture” of the learner is not enough, and the major of the learner should be considered as well. As suggested by Carnegie Mellon University, if a learning session includes students of different majors, it is preferable to either split them into sections or to introduce, for each group, examples that are “relevant to the major or appropriate to the students’ ability” (Carnegie Mellon University, 2015).

Moreover, one of the common questions students ask is “when will I ever use this in the real world?” (Briggs, 2014). Students are usually more concerned in knowing how a certain learning material applies to their major rather than other majors. For example, a civil engineering student uses the probability topic in expecting how much capacity a large container can hold (Prudchenko, 2017) whereas a computer science student uses the same topic in determining how a certain program will act (The University of Chicago, 2017). This is important because learning material should be designed in a way that has a lifelong effect on students and prepares them to their future career as mentioned by Kneale (Kneale, 2009). Several researchers highlight the importance of making learning material relevant to the student’s major. As (Azi et al., 2008) mentions, most people learn by relating the material to what they previously know. Relevance is a key component to motivate learners and help them maintain a good memory of the material they are learning. The neurologist Judy Willis gives an example of memorizing long vocabulary words, she says that if they “don’t have personal relevance or don’t resonate with a topic about which the student has been engaged, they are likely to be blocked by the brain’s affective (or emotional) filters” (Briggs, 2014).

Currently, in multi-disciplinary classes, adapting

Daaboul, T. and Hage, H.
“Majorly Adapted Translator”: Towards Major Adaptation in ITS.
DOI: 10.5220/0007296904510457
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 451-457
ISBN: 978-989-758-350-6

to the major is taken into consideration by human instructors, were some tend to manually modify the same exercise several times, in order to cover all the contexts to suit learners from different majors. Despite helping learners better understand the concepts being taught, this method has several shortcomings: students do not relate to all the examples, it is very tedious, time consuming, and not yet supported by e-learning systems.

The purpose of this paper is to lay the foundations for an ITS, to adapt the topics being taught to the major of the students in an automated and efficient way. For this, we propose "Majorly Adapted Translator (MAT)" an intelligent system that "translates" exercises authored by an instructor from one concept to another while maintaining the same structure of the exercise. This system contains two phases, the first is "Feature Extract (FE)" which identifies the parts of an exercise that need to be translated and adapted to other concepts and the second is "Translate" which is used to translate the exercise into other concepts. This paper highlights the initial phase of the system FE which is based on the Natural Language Processing (NLP) method "Information Extraction". The main purpose of FE is to identify the parts that can be translated and obtain the mathematical form of the exercise to be used in the next phase "Translate" for translating the exercise to other concepts. For this purpose, we have created our own "Relation Extraction (RE)" method that extracts relations of domain specific words by relying on their dependency (grammatical) relations with the rest of the words in the text. Currently, MAT focuses on translating exercises written in human language related to the "Probability" branch of mathematics, since this is a common domain taught to students of different majors such as mathematics, biology, business, computer science, and engineering, to name a few. The paper is organized as follows: the next section provides an overview of CATS, Section 3 details our approach, Section 4 highlights the testing and results, and Section 5 concludes this work and presents the future works.

2 STATE OF ART

Culture or "the programing of the mind" (Hofstede, 1997) affects the way people think, act, and even their understanding of certain matters, which all goes back to what is in their cultural background. In early stages, culture in the subdomains of e-learning was adapted on the level of nation by taking the "human culture perspective" which is based on Hofstede's method who studied the dimensions of culture in organizations from the human's perspective (Hofstede, 1997). These dimensions represent emotions such as pride, teamwork spirit, or ability to accept criticism which are common among certain cultural groups. They are used as the bases for dealing with the learner since they are assumed to be inherited from the learner's cultural group in which he/she unconsciously acts upon when interacting with the system (Blanchard and Frasson, 2005) (Vartak et al., 2008). "Culturally Adapted Tutoring Systems (CATS)" is an example of a system that relies on Hofstede's method. Other than adapting culture in learning techniques, several systems adapted culture in the pedagogical resources given to the learner such as in mathematical tutoring systems (Melis et al., 2009) and authoring tools (Vartak et al., 2008). In later stages, scientists claimed that it is not enough to rely only on the "nation" culture as this results in many "cultural stereotypes" (Ogan et al., 2014). Thus, other factors should be considered as well such as "technological factors" (Nye, 2014), "collaborative filtering" (Eboa et al., 2010) and "Instructional Cultural Contextualization (ICON)" (Mohammed and Mohan, 2014) which learns from analyzing the learner's preferences. Later, (Gasparini et al., 2010) introduced an e-learning system that is more learner-personalized, i.e., it considers factors such as personal, culture, technological, and pedagogical perspective of the user (how much he/she knows about the topic). Currently, many systems adapt the personalization concept in e-learning making a learning session more student-centered such as (Khemaja and Taamallah, 2016) and (Klašnja-Milićević et al., 2017). However, to the best of our knowledge, no work has been done to adapt to the major of the learner.

3 APPROACH

"Majorly Adapted Translator (MAT)" contains two major phases. The first "Feature Extract (FE)" which identifies the parts of the exercise that needs to be translated and adapted to other concepts. The second, "Translate" translates the exercise into other concepts. In this work we highlight the first phase, "Feature Extract". Specifically, Feature Extract defines the structure of an exercise through transforming it into a template. This template includes the list of variables and the mathematical form of the exercise. This template will later be used in order to translate the exercise.

Consider the following sentence: "4% of the resistors are defective". The value 4% should remain unchanged, whereas "resistors are defective" should be contextualized and adapted. We define a variable as "a core part of an exercise that has stable keyword(s) having a related dynamic value which could be changed without affecting the structure of the exercise". A variable is formed of a "keyword" (the "4%" in the previous example) which is an expression that has a mathematical implication depending on the domain of the exercise and a "value" (the "resistors are defective" in the previous example) that refers to all the words linked to the keyword in a way that defines it. The aim is to find the keyword, look for its value and contextualize it into other concepts.

Identifying variables is obtained by extracting the relations of the keywords. However, the challenge is for MAT to be able to find the "correct" value of the keywords since not all terms linked to a certain keyword are its correct value. In this paper, we contributed in creating a "Relation Extraction (RE)" method which extracts relations specific to named entities, by relying on the dependency relations of these named entities with other words in the text. As shown in Figure 1, FE undergoes several steps, detailed in the rest of this section. Note that we consider an exercise as composed of two parts: the *given* where all the details are provided, and the *questions* that the learner must answer.

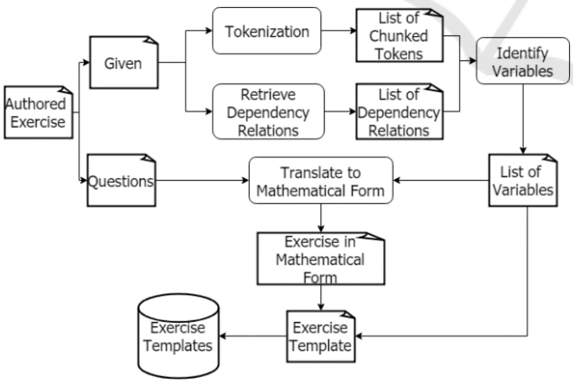

Figure 1: Feature Extraction Process.

3.1 Tokenization

The initial step in FE is *"Tokenization"*; it divides the *"given"* of the exercise into *"chunks of words/tokens"* which are a group of words belonging to the same grammatical type. FE identifies chunks as either a "keyword chunk" or a "non-keyword chunk" and labels each with its grammatical type which is used as a key to identify the relations among the chunks. Like any other Information Extraction (IE) task, the initial step is to split the text into smaller parts, i.e., splitting the given into sentences and then to clauses in which each undergoes the process of identifying the variables. Splitting into clauses is helpful since the smaller the text, the easier it is for the dependency parser, used in the next step, to extract accurate dependency relations. Next each clause is divided into tokens using a "word tokenizer" and the part-of-speech tag of each token is retrieved which will be used to group them into chunks.

Next, the Stanford NER (Named Entity Recognition) tool is used to identify the "Keyword Chunks". Stanford NER is a statistical parser created by Stanford University, trained on previously annotated text that labels sequence of words referring to "name of something" (Manning et al., 2014) (Finkel et al., 2005). Since NER is a domain specific tool, it was required that we customize the Stanford NER to identify keywords related to the domain of MAT, i.e., mathematical or probability expressions. Stanford NER can identify numbers such as percent, integer, and money. Whereas other types of numbers and mathematical/probability expressions that Stanford NER cannot read, are identified using "Stanford Regex NER"; a rule based NER system that uses Java regular expressions in order to label named entities. Accordingly, we have trained the Regex NER to identify specific probability expressions which we created by developing our own Java regular expressions such as: "2 out of five" is identified as "2/5", and the words "probability, random, standard deviation, average, median, etc." are identified as keywords.

Furthermore, all the chunks that are not identified as keyword chunks are considered non-keyword chunks and parsed using a shallow parser. Chunking non-keywords at early stages is important because it simplifies the work of analyzing the relations between a keyword and its value. Consider the phrase "20 have a laptop computer" where "20" is the keyword; instead of obtaining the relations: "20" linked to "laptop" and "laptop" linked to "computer"; having "laptop computer" as one chunk of type "Noun Phrase", directly leads to the relation that "20" is linked to "laptop computer". In order to obtain similar forms of chunks such as noun phrases and verb phrases, we developed our own grammar customized for MAT to perform the shallow parsing. This grammar contains "tag patterns" which were derived from studying various grammatical rules and from looking at frequently encountered chunks related to a value of a variable found in mathematical text. Finally, we reserved for each chunk, its type, the original token value, its normalized form, and most importantly the index of each token in the chunk in order to link it later on with the indexes of the dependency relations.

3.2 Retrieve Dependency Relations

After identifying the keyword chunks and the non-keyword chunks, the second step is to extract the dependency relations among these chunks. This step relies on the Stanford Core NLP dependency parser (Manning et al., 2014). It uses its “enhanced” dependency relations that consists of a governor (or the head) and a dependent that depends on the governor. In order to analyze the dependency relations between chunks, we studied various dependency type relations. One of the most important relation types to mention is the “core argument” type which is a relation that includes a subject and an object (Universal Dependencies Contributors, 2016). Consider the phrase “three have defects”, the following relations are obtained: “three” is the subject of “have” represented as *nsubj(have,three)* and “defects” is the object of *“have”* represented as *dobj(have,defects)*. This step helps the system in the next stage to link certain chunks to the keywords according to their dependency relations, forming up a “variable”.

3.3 Identify Variables

This step is the most important step in FE, as it detects the parts of an exercise that needs to be translated into other concepts, i.e., the *“variables”*. “Identify Variables” takes as an input the list of chunked phrases obtained from “Tokenization” and the list of dependency relations determined by the “Retrieve Dependency Relations” step. As mentioned previously, not all relations of a keyword, are supposedly its value. Hence, the challenge is to obtain the correct value of the variables; FE performs this through three main steps:

First, “*Getting the linked Relations*” extracts all the linked relations of the keyword(s) and omits those that are not. A chunk can be directly linked to a keyword chunk or indirectly through other chunks linked to the ones related to the keyword. Moreover, a value can be found in more than one chunk, so the dependency relations among these chunks are to be considered as well. All the dependency relations obtained from this step are called “linked relations”. Moreover, MAT considers a predefined priority list of keywords; this list was carefully created based on scanning similar examples, such that it gives the shortest path leading to the value and accurate results. Accordingly, the system starts with the priority keyword, gets all its directly linked relations and then completes to get the relations linked to the direct chunks, i.e., the indirect relations. This process stops when the system cannot find linked relations any further. Consider the phrase “Eleven of the motors are free of defects” where “eleven” is a keyword of type “Number” and “free” is a keyword of type “Negation”. According to the predefined priority list, MAT starts with “eleven” and obtains the direct relations: “eleven” is the nominal of “motors” and “free” is the subject of “eleven”. Since “free of defects” is one chunk, thus “eleven” is linked to “free of defects”. The indirect relation obtained is that the verb “are” is the copula that connects to the subject “free” and “free” is the nominal of “defects”.

Second is the “*Translating Relations*” step which helps the system understand the relations among the dependencies and how to use them as instructions to identify the variables. It does that by translating the linked relations based on “the type of relation” and on the “type of the related chunks”, as well as eliminating relations that are considered useless. In the previously mentioned example, *“Eleven of the motors are free of defects”* the obtained dependency relations are translated as:

1. *“nmod:of(11,motors)”* ➔ *“Number is NounPhrase”*
2. *“nsubj(11,free)”* ➔ *“Number has Subject Negation”*
3. *“cop(free,are)”* ➔ *“Negation has Equality”*
4. *“nmod:of(free,defects)”* ➔ *“Negation has NounPhrase”*

Third, “*Variable Identification”* aims to identify the correct value of a keyword by understanding the translated linked relations obtained from the previous steps. FE defines an algorithm that extracts the “value” for different types of keywords. Essentially, while searching for the “value”, the system undergoes two levels of search, the first searches in the direct relations and the second in the indirect relations. Throughout both levels of search, the system performs three actions: “Search”, “Continue”, and “Add”. Based on the type of the related chunks and the type of the relation, the system decides whether to add the word as a value or ignore it and continue the search. As the system analyzes the dependency relations, it relies on general grammatical rules inspired from previously chunked variables of similar structure. For example, when a keyword is linked to a "verb" by the relation "subject", the system understands that there should be an "object" so it continues to search for it, if the encountered object is a noun, then it is added to the "value". In the previously mentioned example, the keyword "11" is linked to a negation "free" by the relation subject, so FE adds "free" to the value and continues to add the chunks related to "free" as part of the value, forming up the value "free of defects" for the keyword "11". Furthermore, when MAT identifies the value, it defines the type of it. A value can be a regular value type (denoted as "Value") or of the types: "Key Is", "Number", or "Relation (i.e., negation or intersection)" type. According to the type of the value, FE determines the variables. In the above mentioned translated results, FE identifies "motors" and "are free of defects" as values of "11" of value type "Key Is" and "negation" respectively. Thus, the obtained variable is "11 (of the motors) =free of defects".

3.4 Translate into Mathematical Form

The final step, "Translate into Mathematical Form" generates the mathematical form of the exercise. First, it translates the extracted variables, then the given part of the authored exercise, and finally the questions associated with the exercise. Basically, MAT tries to understand the given as if trying to solve it, taking advantage of the structure of the variables and related mathematical rules. As probability exercises vary in type and in the way they are solved, MAT defines different algorithms for different categories of probability exercises. For instance, for probability exercises of type "basic probability", the system searches for the "sample space", the events and the possible negation or intersection relations. Following are examples of obtained variables: (1): 20.0 (electric motors); (2): 11.0 (of the motors) = free of defects; (3): 8.0 = have defects on the exterior finish; (4): 3.0 = have defects in their assembly. MAT "*translates*" (3) and (4) as events A and B because they are related to the keyword with the value type "value"; (2) is identified as a "relation value" since it has the value negation (Not A and Not B), and (1) is the sample space. After translating the variables, MAT displays the given in mathematical form using predefined probability symbols. For example, it associates each event with its number value, e.g., Event A= 8.0. Also, it identifies terms that have mathematical implications, e.g., the word "at least one" implies a union P(AUB). Thus, the translated mathematical form is displayed as: *"Consider the Sample Space 20.0, let the events A and B, 8.0 is the number having A, 3.0 is the number having B, 11.0 is the number having NOT A & NOT B."* Finally, MAT translates the questions associated with the exercise to mathematical form using the same approach.

4 TESTING AND RESULTS

The performance of Feature Extract (FE) is evaluated based on the percentage of exercises it can produce a valid template for and the various types of probability exercises it can cover. Based on the recommendation of statistics and probability professors, and after researching commonly used statistics/probability books (such as (Scheaffer et al., 2010) and (Grinstead and Snell, 2009)) and other online resources, we decided to cover the following topics for the testing: Basic Probability (20 questions), Permutation & Combination (24 questions), Conditional Probability (14 questions) and Distributed Probability (12 questions). The reason behind choosing these categories is that they are the most common and preliminary to several majors requiring probability courses. As such, a total of 70 different exercises were selected to cover the various question formats and topics. The exercises were first translated to mathematical form using MAT. Then, with the help of probability instructors, the translated forms were compared and evaluated either as correctly translated, partially translated (containing some minor errors) or incorrectly translated. Overall, MAT succeeded in extracting the templates and translating correctly to the mathematical form 84% of the exercises. Moreover, MAT was able to partially translate 14% of the exercises and incorrectly translated only 2% of the exercises.

4.1 Results and Findings

Figure 2, shows the detailed results of each tested category. As observed, FE performed well in translating correctly exercises of the different categories. As for the exercises that were partially and incorrectly translated, we observed several issues in which the majority of them could be fixed.

First of all, in the "Basic Probability" category, MAT correctly extracted and translated 90% of the exercises and it performed well in identifying events and sample spaces, and distinguishing between different probabilities symbolic forms.

As for the "Conditional Probability" category, in 86%

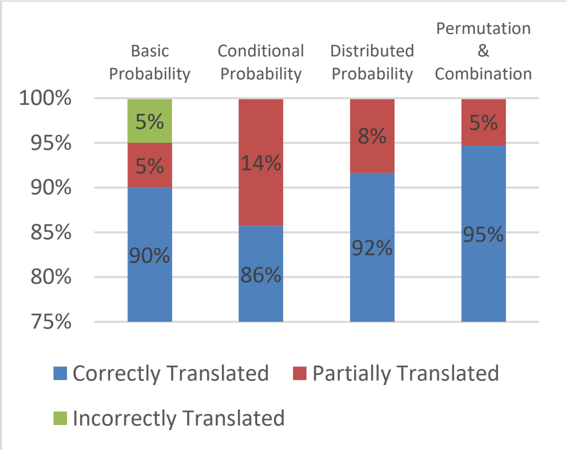

Figure 2: Detailed Results.

of the exercise FE was able to detect conditional probabilities in the form of P(A|B) whereas in 14% of the exercise, where the given was more complex, it was only able to translate it partially. In the "Distributed Probability" FE also performed well, as it was able to correctly translate 92% of the exercises. Finally, in the "Permutation and Combination" category FE was able to correctly translate 95% of the exercises.

The shortcomings of the FE component were due mainly to the following reasons. First, FE assumed that all the variables were in the given. In the cases, where not all the events were presented in the given, but some in the questions, MAT incorrectly translated the exercises. This issue can be solved by performing further text analysis and extracting the keywords that were not found in the given. Second, FE had some difficulties in exercises having unidentified keywords such as *"X members attend Sport"*. This can be solved by further extending the FE to recognize single letters to be keywords. Finally, FE failed to identify "list of words" which infers a number, for example the phrase "*the vehicle can go straight, turn right, or turn left"* which infers 3 choices, denoted as $n=3$. This can be solved by training FE to identify list of consecutive words separated by a comma as "values".

Overall, the performance of FE is considered to be acceptable, especially that the FE phase can successfully handle complex terms including numbers ("*two out of five"* is recognized as 2/5), units (such as *"10 ohm resistors"*), negations (such as "*neither*", "*nor*" or "*free of*") and other ambiguous terms ("*A+*" is recognized as a blood type).

5 CONCLUSION AND FUTURE WORKS

This paper contributes first in providing arguments in order to expand the field of AIED towards adapting to the major of the learner. The proposed system, "Majorly Adapted Translator" (MAT) is designed for that end. Indeed, MAT adapts to the students' major by translating exercises from one concept to another according to their major. The system consists of two parts, "Feature Extract" which identifies the parts of an authored exercise that must be changed (i.e., variables) and "Translate" which translates these to different concepts. In this work, we highlight the Feature Extract phase, which relies on our own relation extraction method to identify variables which extracts relations specific to named entities by using dependency relations and shallow parsing.

The system was tested on 70 different exercises, which were selected to cover the various question formats and topics from the Statistics and Probability domain. MAT was successful at properly extracting the templates and translating into mathematical form 84% of the exercises. Moreover, in 14% of the exercises MAT was partially successful, and the reasons for these limitations were determined.

As future works, first we plan to increase the performance of the FE phase by addressing the limitations highlighted in the testing section. Second, the translate phase only translates to mathematical form. As such, it should be extended to translate other majors in order to further test the system with actual learner. In addition, we plan to investigate extending the system to translate various topics other than probability such as linear programming problems and numerical analysis.

REFERENCES

Azi, J. I., Aboderin, A. O., Nkom, A. A., and Schweppe, M., 2008. Dialogue on Integrating Indigenous Contents and Instructional Technology Application in Nigeria: The "i-CLAP" Model Experience. In *EMERGE'08, Emerge Conference*, South Africa.

Blanchard, E. G. and Frasson, C., 2005. Making Intelligent Tutoring Systems culturally aware: The use of Hofstede's cultural dimensions. In *IC-AI'05, International Conference on Artificial Intelligence*, Las Vegas, pp. 644-649.

Briggs, S., 2014, *How To Make Learning Relevant To Your Students (And Why It’s Crucial To Their Success)*, viewed 23 September 2018, <https://www.opencolleges.edu.au/informed/features/how-to-make-learning-relevant/>.

Carnegie Mellon University, 2015. *Solve a Problem: The course draws students across majors*, viewed 20 September 2016, <https://www.cmu.edu/teaching/solveproblem/strat-backgroundknowledge/backgroundknowledge-02.html>.

Eboa, F. M., Courtemanche, F., and Aïmeur, E., 2010. CAMPERE: Cultural Adaptation Methodology for Pedagogical Resources in E-learning. In *CATS'10, Third International Workshop on Culturally-Aware Tutoring Systems*, Montreal, pp. 37.

Finkel, J. R., Grenager, T., and Manning, C., 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, In *43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, pp. 363-370.

Gasparini, I., Bouzeghoub, A., De Oliveira, J. P. M., and Pimenta, M. S., 2010. An adaptive e-learning environment based on user´s context. *In CATS'10, Third International Workshop on Culturally-Aware Tutoring Systems*, Montreal, pp. 1 -12.

Grinstead, C. and Snell, J., 2009, *Introduction to Probability.* American Mathematical Society.

Hofstede, G., 1997 *Cultures and Organizations: Software of the Mind*, McGraw-Hill, London.

Khemaja, M. and Taamallah, A., 2016. Towards Situation Driven Mobile Tutoring System for Learning Languages and Communication Skills: Application to Users with Specific Needs. *Journal of Educational Technology and Society*, vol. 19, no. 1, pp. 113-128.

Klašnja-Milićević, A., Vesin, B., Ivanovic, M., Budimac, Z., and Jain, L. C., 2017. *E-Learning Systems-Intelligent Techniques for Personalization*, Springer,Cham.

Kneale, P., 2009. Teaching and Learning for Employability. In Fry. H., Ketteridge, S., & Marshall, S. (eds.), *A Handbook for Teaching and Learning in Higher Education*, 3rd edition, Routledge, NewYork, pp. 99-112.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

Melis, E., Goguadze, G., Libbrecht, P., and Ullrich, C., 2009. Culturally adapted mathematics education with ACTIVEMATH. *AI and Soc*, vol. 24, no. 3, pp. 251–265.

Mohammed, P. and Mohan, P., 2014. Dynamic contextualization of educational context in intelligent learning environments using ICON. *International Journal of Artificial Intelligence in Education*, vol. 25, no. 2, pp. 249–270.

Nye, B., 2014. Intelligent Tutoring Systems by and for the Developing World: A Review of Trends and Approaches for Educational Technology in a Global Context. *International Journal of Artificial Intelligence in Education*, vol. 25, pp. 177–203.

Ogan, A., Walker, E., Baker, R., T. Rodrigo, M., Soriano, J., and Castro, M., 2014. Towards understanding how to assess help-seeking behavior across cultures. *International Journal of Artificial Intelligence in Education*, vol. 25, pp. 229–248.

Prudchenko, K., 2017, What Kind of Math Is Expected of a Civil Engineering Student?, viewed 23 May 2017, <http://education.seattlepi.com/kind-math-expected-civil-engineering-student-2159.html>.

Scheaffer, R., Mulekar, M., and McClave, J., 2010. *Probability and Statistics for Engineers*. 5th Edition, Cengage Learning.

The University of Chicago, 2017, *Math Needed for Computer Science*, viewed 23 May 2017, <https://csmasters.uchicago.edu/page/math-needed-computer-science>.

Universal Dependencies Contributors, 2016, *Universal Dependencies v2*, viewed 20 January 2017, <http://universaldependencies.org/>.

Vartak, M. P., Almeida, S. F., and Heffernan, N. F., 2008. Extending ITS Authoring Tools to be Culturally Aware. *In CATS'08, Culturally Aware Tutoring Systems*, Montreal, pp. 101-105.