A Novel Framework for Math Word Problem Solving

Kyle Morton and Yanzhen Qu

Abstract—**Mathematical word problems represent many real world scenarios. Often times found difficult to solve. The reason for their difficulty is because of miscomprehension and the inability to formulate a mathematical representation of the text. In this paper we present a framework based on fuzzy logic – ontology model that interprets a mathematical word problem in natural language and compute a solution. This framework uses ontology as a working memory and search engine to compute the solution. Fuzzy logic is used to determine the confidence of result returned, which could eliminate the confusion for a user trying to determine if the solution provided is correct. The ability to interpret a mathematical word problem and return a detailed solution will help educated users by providing them detailed steps of a solution.**

Index Terms—**Ontology, fuzzy logic, artificial intelligence, education, e-learning, word computing.**

I. Introduction

The computation of mathematical word problems opens a domain of real world solutions. The ability to formalize a word problem in natural language and process it provides a user interface that is easy to learn, operate, and encouraging to use. Currently mathematics software interfaces remain clumsy and non-user-friendly. Even though the provision of a standard protocol and syntax for mathematical input is a remote possibility, users often feel reluctant to learn yet another syntactic convention [1]. As for children and adults, people are most challenged by word problem solving not because of their mathematical skills but because of text comprehension. Regularly, incorrect answers to word problems are because of correct calculations to incorrect problem representation [2]. Current search engines cannot solve mathematical word problems, if a user wanted to query the solution to a math problem traditional search engines will only return the calculated result. Only returning the result cripples the learning ability of users because he or she is not learning how to solve the problem. The purpose of this paper is to introduce a fuzzy logic – ontology model that is geared to use natural language processing (NLP) to interpret text and Wolfram|Alpha which is a search engine that incorporates mathematica to solve mathematical equations, consequently educating users by providing supporting detailed steps of the solution.

This paper is structured as follows. Section II presents an extensive literature review on the attempts to create a search engine model that promotes user learning while searching the Web and made attempts to create a model to solve mathematical word problems. Section III is the problem statement and hypothesis. Section IV presents our model which contains the components of ontology and fuzzy logic. Section V presents an example of applying our framework. Last, Section VI offers the conclusion.

Manuscript received November 13, 2012; revised December 12, 2012.

The authors are with School of Computer Science Colorado Technical University Colorado Springs, USA (email: k.morton15@my.cs.coloradotech.edu, yqu@coloradotech.edu).

II. Related Work

A. Solving Word Problems

Many people find word problems difficult to solve. According to [3], students perform 30% worse on arithmetic word problems than on problems in numeric form. Statistics show that students do not have a problem solving a numerical equation but have difficulty comprehending word problems. If a person does not have the knowledge to comprehend and solve word problems in mathematical situations that person is challenged with building a semantic relationship of the data in different word problem types. Researchers stop short defining the problem of solving word problems with the inability to relate working memory and the ability to process of "translating" natural language into correct mathematical relationships or into solutions [4]. Cummins *et al.* [3] state that there are two explanations why word problems are troublesome, the lack of logico-mathematicaldevelopment and the lack of linguistic development. The logico-mathematical development view explains that people fail to solve certain problems because they lack the conceptual knowledge to solve them. The linguistic development view explains that word problems are difficult to solve because they have linguistic forms that do not map onto a person's existing conceptual knowledge [3].

These two views have a role in how people solve certain word problems of semantic structure. These semantic structures can be classified in three types: combine, change, and compare [2]. The combine structure is defined as sets that are combined (e.g., "John has 2 cars. Mark has 1 truck. How many vehicles do they have total?"). Change is defined as a set which changes over time (e.g., "James has $3. He then found $5. How much money does he have now?"), and a compare structure is the comparison between two sets (e.g., "Rick has 2 kids. Mike has 3 kids. How many more kids does Mike have?"). Compare problems are more difficult than the other types because they describe static relationship and they are the only type that incorporates relational terminology [2]. The framework we are presenting in this paper goes beyond the three semantic structure types. It uses ontology as a working memory and fuzzy logic in determining what operation needs to be performed. This approach allows the interpretation of a word problem, build data relationships, and apply the correct operation to compute a word problem.

B. Word Problem Simulation Models

There have been many mathematical word problem simulations developed to study the affect word problems have on children. In order for a simulation to account for potential errors made by people, "Asimulation should account for misinterpretations (e.g., "some" could be a qualitative adjective), must provide access to individual lexical entries, and must be sensitive to a problem's wording" [5].

LeBlanc and & Weber-Russell's word problem simulation [4] analyzes the difficulty solving word problems as wording in the problem changed. They used a bottom-up approach to simulate young and novice word problem solvers. They define "novice" as one who reads a sentence and then tries to integrate the new information before proceeding to the next sentence. Their simulation is based on two components, EDUCE and SELAH. EDUCE is a parser that is distinct for word problems. It focuses on quantities, noun compounds, pronominal reference, time-sequence, set partitions, ellipsis, and references to previous sets. SELAH is a text integration which is used as a tool to integrate information that is in working memory represented by EDUCE. The weakness of their model is that it is designed to simulate novice word problem solvers. This is a problem because a bottom-up approach does not allow you to build proper data relationships on prior knowledge. Also their model does not comprehend the relationship between data sets and objects, only allowing their model to compare, combine, and change data to compute a solution. Our model uses the data associated with specific objects given by an ontology to build a mathematical equation which is computed.

According to [4], "Reusser's Situational Problem Solver possesses an elaborate text-processing component to distinguish the representation of the situation from the representation of the text. Reusser's is the first attempt to simulate the progressive and incremental process of transformation from text to situation to equation. In particular, SPS stresses that from a problem-solving (and instructional!) point of view, arriving at a representation of the situation in a problem is not a superfluous process, but rather a necessary one. Although SPS reads and solves a wide range of very complex problems involving active descriptions of changes in quantities, it does not address static and/or relational language problems (e.g., Combine and Compare types)." Using ontology our model is able to handle language relationships, such as children infer boys and girls, unlike Reusser's model. Briars & Larkin simulation "CHIPS" solves word problems using physical counters that represent objects [6]. For example, "the sentence "David has 5 baseball cards" causes CHIPS to put out five counters, each labeled as a baseball card belonging to David. If this sentence were followed by: "Then Pat gave 2 baseball cards to David," CHIPS would move two new counters into David's pile, update these counters as belonging to David, and make one pile of seven counters" [4]. Our model uses ontology as a working memory for sets and their relationships which differs from CHIPS counter model. Note that the aspects of Cummins's model is that it solves problems through an interaction of text-comprehension processes and arithmetic problem solving strategies [3]. Our model uses numerical information as a set of objects but do not comprehend text using proposition frames like Cummins's model does.

[7] use natural language processing steps of morphological, syntax, and semantic analysis to make implications of word problems. To represent the interpreted mathematical structure and change it into a documental metadata they used Mathematical Makeup Language (MathML), which is also used by [8] and Scalable Vector Graphics (SVG), within the eXtensible Makeup Language (XML). To understand phrases and sentences Chinese Broken Interrupt Phrase (CKIP) was used to mark the sentence structure which inconsistently does not provide reasoning of a phrase. To fix this issue they omitted certain parts of speech to decrease the difficulty to change a phrase into a more meaningful expression. Our model uses the Tagging of the NLTK to mark a phrase and then creating an ontology from the relationship of the data. Their model also matches and classifies each mathematical condition to a rule base which triggers the execution of a rule when satisfied. The issue with this approach is that some word problems may be misinterpreted and an incorrect rule may be applied, which would result in incorrect solutions.

To solve multi-step addition and subtraction word problems[9] proposes MSWPAS. MSWPASconsists of usingnatural language processing to comprehend word problems and then constructing problem frames to store data. When comprehending they categorize each proposition into a slot consisting of object, number, and specification and role, and time. Each frame represents a slot and the time stamp dictates the order of execution. The downside to their model is that they focus on mathematical word problem types of change, combine, and compare. Their model is also limited to addition and subtraction problems.

Our model uses Wolfram|Alpha for querying an auto-generated equation for a solution. [10] uses a similar approach but consists of retrieving mathematical documents using Kohonen's Self-Organizing Map (SOM) for data clustering of similar mathematical documents from a mathematical document database, which could be created manually or downloaded from the Internet [10]. The purpose of these documents is only to provide hints or possible solutions to their query. This could be misleading to a student because hints provided by a search engine may not be enough for a person to learn how to solve a mathematical problem. The query results given from our model provides a step-by-step solution to a problem which allows a person to correct their mistakes and also teach them the proper steps to solving a problem.

III. Problem Statement and Hypothesis

A. Problem Statement

Currently, to any inquiry of math word problem, all search engines merely return lists of web documents as results of key word match, which vary in correctness or relevancy, resulting in a tedious selection process.

B. Hypothesis

A framework based on fuzzy logic – ontology modelto solve math word problem will enable a search engine to return results with greatly improved correctness and relevancy for any math word problem inquiry.

IV. Methodology

The components of this framework are natural language processing, ontology, fuzzy logic, and querying a search engine for a solution. These components together allow this framework to perform a more effective search of a solution for an equation representing a word problem.

First this framework begins natural language processing a question, phrase, or story. This process consists of analyzing each sentence and determining the relationship between objects by creating an ontology as a working memory. Once relationships has been created and objects has been formed from data in the working memory categories that correspond to word problem types are weighted to determine which category is closest related to the word problem. Using the category that is selected, an equation is created using the information from working memory. After creation of the equation the equation created is weighed. The equation is next queried for a solution. The strength of the result is then determined using fuzzy logic. Algorithmically we can abstract the framework in the following pseudo code:

```
Interpret the context of the word problem
Build Ontology
SET highest_weight TO 0
SET highest_category TO 0
FOR ( i = 1; i <category_count; ++i; ) DO
        weight = compute category[i] weight
        IF weight is >highest_weight
                SET highest_weight TO weight
                SET highest_category TO i
        END-IF
END
Create Equation using highest_category
Determine weight of equation
SET equation_weight TO weight of equation
Query Equation
Determine confidence of result using highest_weight and equation_weight as membership values
```

A. Natural Language Processing

Natural language processing (NLP) is used to analyze the context of a sentence. When a user submits text to query our systems begins tagging parts of speech and building semantic relationships. For example if a user enters, “Mike has 5 marbles. Chris has 6 more marbles than Mike. How many marbles does Chris have?” Our system interprets Mike as a person that has five marbles and Chris who has six marbles, then the last sentence as a possibility of how to compute the context. These sentences are parsed and converted into objects which are represented in an ontology. The ontology is then used to build a search query. Our NLP processor is built in Python using the Natural Language Tool Kit (NLTK) [11] which is a framework used for part-of-speech tagging, syntactic parsing, and text classification. NLTK provides the ability to solve complex word problems by assisting in building an extensive working memory. In terms of mathematical word problems our NLP builds an equation based on the knowledge in working memory. Based on the keywords defined by the NLP we next determine the relevance of the keywords to a word problem category type. In this paper we will focus on four types of word problems, which are investment, distance, projectile, and percent. Investment word problems are problems that involve simple or compound interest. Simple interest uses the formula $I = PrT$, where I stand for interest on the original investment, *P* (principal) stands for the amount of the original investment, *r* is the interest rate, and *t* is the time. Compound interest uses the formula $A = P\left(1+\frac{r}{n}\right)$, where A is the ending amount, P isthe beginning amount, r is the interest rate, *n* is the number of times it compounds a year, and *t* is the total number of years. Distance word problems involves an object(s) traveling at a fixed or average rate determining how far, fast, or long. It uses the formula $d = rt$, where *d* stand for distance, *r* is the rate of speed, and *t* stands for time. Projectile word problems consists of objects being thrown, shot, or dropped using the formula $s(t) = gt^2 + v_0t + h_0$, where *g* is the force of gravity, v_0 is the initial velocity, h_0 is the initial height, and *t* stands for time. Lastly, percent word problems involve computing the percentage. These categories have defined terms which resembles to the context of a word problem.

Each category is weighed to determine which is most relevant to the context of the word problem. This allows the system to know which formula to use. Categories are ranked using (1) which is derived from a semantic term weighting model[5]. Their approach is based on the intuition that the importance of a term to a document is dependent on its frequency as well as the degree of rareness at the document level in the corpus [12].

$$TF-IDF\left(t_i, d_j\right) = \frac{\sum_{i=1}^{c} count\left(t_i, d_j\right) \times |\log \frac{corpus\text{賤}count\_doc(t_i, corpus)}{count\_doc(t_i, corpus)}|}{corpus} \quad (1)$$

where count (t_i, d_j) refers to the frequency of term t_i in document d_j, also known as term frequency (*tf*); corpus refers to the number of documents in the corpus; *c* refers to the total number for terms in a query; count_doc (*t*, corpus) refers to the number of documents in the corpus that contains a term in set *t*.

Once a category is selected a formula is built by the NLP using the ontology. The weighting of a mathematical equation built by the NLP is determined by what type of mathematical operation is being performed, the current information (*ci*), and the information required (*ir*) for completion of an equation. The *ci* is the information available that is used to complete an equation. The *ir* is the information required to compute an equation. In the example, "Mike has 5 marbles. Chris has 6 more marbles than Mike. How many marbles does Chris have?" The equation $z = x + y$, represents what is required which is *x* and *y*. The *ci* is Mike's five marbles and Chris's six more marbles, which is represented by $z = 5 + 6$. To compute the weight of a math equation we use (2).

$$weight = \frac{ci}{ir} \tag{2}$$

B. Searching and Weighting Results

Our framework uses Wolfram|Alpha as a search engine to compute mathematical equations. Wolfram|Alpha is a web application that does dynamic computations based on data it has stored and the corresponding algorithms. It provides factual answers to queries and also provides facts that complement the answer and how the answer was derived. Most importantly it uses Mathematica which is computational software used in engineering and mathematics.

C. Ontology

An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them[13]-[14]. The importance of building and using ontologies allows us to associate information from the context of the word problem to formalize an equation. From an user's perspective, in adaptive learning systems, ontologies are exploited not only to organize learning objects and to state their inter-relationships but also to build personalized e-Learning experiences and to maintain up to date students cognitive states [15]. Ontologies allow us to derive and investigate the reasoning of an object.

This model uses a semantic mark-up language referred to as "Web Ontology Language" (OWL), which is a vocabulary extension of Resource Description Format (RDF). RDF is a flexible approach to represent data that can be used to define context in web documents. OWL adds more reasoning beyond the schema of RDF which contributes to more describing properties and classes for data and relational-data. Both OWL and RDF are built upon the surface and schema of XML and also extending the XML data types [16].

D. Fuzzy Logic

Results are possible solutions to the equations queried by the search engine. After the results have been retrieved, fuzzy logic is used to determine the confidence of the result returned. Fuzzy logic provides validity for answers returned by the search engine based on the membership value of the category and equation associated to the word problem. Fuzzy rules govern the fuzzy system allowing it to use the information in a deterministic way. A fuzzy rule consists of an if-part (antecedent) and a then-part (consequence). In simpler terms an antecedent describes a condition and the consequence describes a conclusion. The fuzzy rules used in this paper are defined in Table I.

TABLE I: FUZZY RULES

Rule	Antecedent	Operator	Antecedent	Consequence
R1	Category is Low	AND	Equation is Low	Result is low
R2	Category is Low	AND	Equation is Medium	Result is low
R3	Category is Low	AND	Equation is High	Result is low
R4	Category is Medium	AND	Equation is Low	Result is low
R5	Category is Medium	AND	Equation is Medium	Result is medium
R6	Category is Medium	AND	Equation is High	Result is medium
R7	Category is High	AND	Equation is Low	Result is Medium
R8	Category is High	AND	Equation is Medium	Result is High
R9	Category is High	AND	Equation is High	Result is High

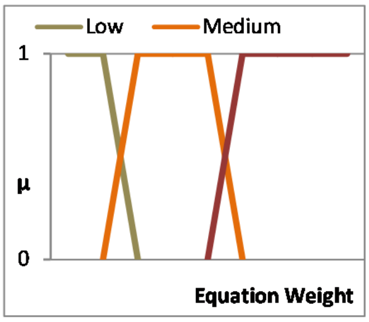


Fig. 1. Membership function for weight category

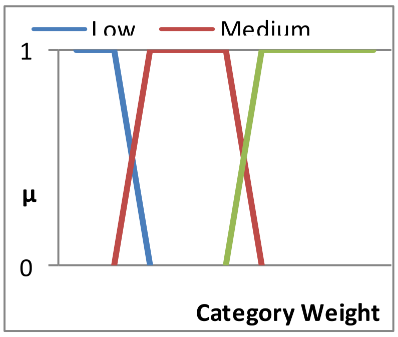


Fig. 2. Membership function for equation weight

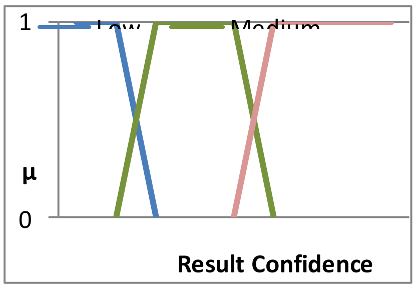


Fig. 3. Membership function for result confidence

In this paper we let the membership value of a category be represented by the weight that is computed by (1), using the membership function in Fig. 1.The membership value of an equation produced by the NLP after a category has been selected is computed using (2), which uses the membership function in Fig. 2. After the fuzzy matching of the (1) and (2) with their appropriate membership function the inference step is invoked for each of the relevant rules to produce a conclusion based on their matching degree, in which this combination of conclusions uses the membership function is Fig. 3. These fuzzy rules has multiple conditions which are combined using AND (conjunction) and a min fuzzy conjunction operator. The min defines the conjunction of two fuzzy sets as the minimal of degrees of membership of the two fuzzy sets, which is also explained as $\mu_{A\wedge B}(x) := \min\{\mu_A(x), \mu_B(x)\}$ .Where the OR (disjunction) represents the max of degrees of membership of two sets, also explained as $\mu_{A\vee B}(x) := \max\{\mu_A(x), \mu_B(x)\}$.

V. Application of the Framework

To demonstrate our framework, we have applied it to a mathematical word problem. A *"distance"* word problem is used to address the relationships between different objects and data. The word problem for this example is:

Two cars started at the same time from opposite ends of a road that is 45 miles long. One car is driving 55 mph and the second car is riding at 65 mph. How long after they begin will they met?

The model begins natural language processing the text. This process consists of analyzing each sentence and determining the relationship between objects by creating an ontology as a working memory. The ontology of the example is shown in Fig. 4.

The ontology in Fig. 4 consists of two classes which are a *car and road* class and two datatype properties which are *mph and miles*. An OWL class provides an abstract grouping of similar characteristics. The classes contain a comment property which is a human readable description about the resource and a label property which is a human readable version of a resource's name. A datatype property relates individuals of classes to literal values. Individuals which identifies a resource and contains elements describing a resource, contains information about the cars' rate of speed and distance traveled. Based on the context of the example this word problem can be categorized as a problem involving distance. To determine what mathematical equation to use for this word problem we use (1) to determine which keywords from the word problem best corresponds to the appropriate category. Eq. 1 allows us to determine the word relevance of a certain category in respect to keyword frequently and inverse keyword frequently, which favors terms concentrated in few documents [5]. In this example we use four types of word problems which are investment, distance, projectile motion, and problems computing percentages. Each type of word problem has a category with conforming keywords that are used as keyword documents to serve as the corpus in (1). Table II demonstrates how each category weighs after determining the relevancy of keywords from the word problem.

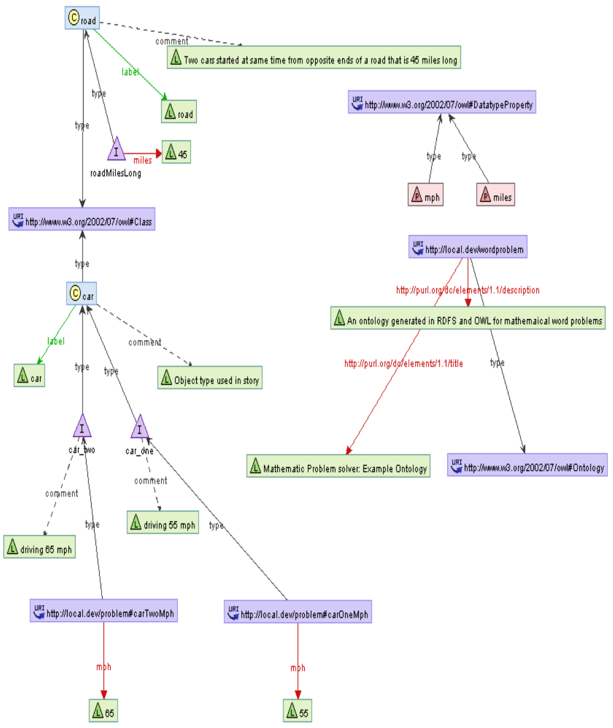

road
comment
Two cars started at same time from opposite ends of a road that is 45 miles long
label
type
road
type
miles
45
roadMilesLong
http://www.w3.org/2002/07/owl#DatatypeProperty
type
type
mph
miles
http://www.w3.org/2002/07/owl#Class
http://local.dev/wordproblem
http://purl.org/dc/elements/1.1/description
An ontology generated in RDFS and OWL for mathemaical word problems
type
car
label
comment
car
type
type
Object type used in story
http://purl.org/dc/elements/1.1/title
type
car_two
car_one
Mathematic Problem solver: Example Ontology
http://www.w3.org/2002/07/owl#Ontology
comment
comment
driving 55 mph
type
type
driving 65 mph
http://local.dev/problem#carTwoMph
http://local.dev/problem#carOneMph
mph
mph
65
55
Fig. 4. Ontology of working memory

TABLE II: Category Weights

Category	*tf*	TF-IDF (Weight)
Investment	0	0.0
Distance	6	0.7
Projectile	2	0.2
Percent	1	0.1

Next a mathematical equation is formed from the data in the working memory, which uses the formula $d = rt + rt$. A representation of the formula is displayed in (3).

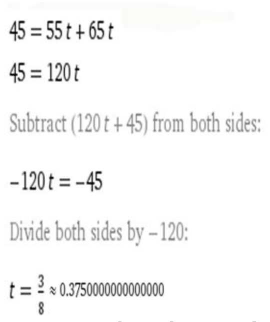
$$45 = 55t + 65t \tag{3}$$

Once the data that is needed and the data that is present to satisfy the formula are determined we use (2) to compute the weight of the equation, as detailed below.

$$\mu(x) = \frac{ci}{ir}$$

$$\mu(x) = \frac{3}{5}$$

$$\mu(x) = 0.6$$



$45 = 55t + 65t$

$45 = 120t$

Subtract $(120t + 45)$ from both sides:

$-120t = -45$

Divide both sides by -120:

$t = \frac{3}{8} \approx 0.3750000000000000$

Fig. 5. Search result returned

The equation $45 = 55t + 65t$, for t is then queried using

Wolfram|Alpha search engine. The search result is then returned, shown below in Fig. 5.

This result has a high confidence value based on fuzzy rule R8 in Table I. Fig. 6 explains that the degree of membership for the category was 0.7 (high) and the degree of membership for the equation was 0.6 (medium) concludes that the degree of the input that matches the fuzzy rule is 0.6 (high), which entails that computed result has a high confidence level of being correct.

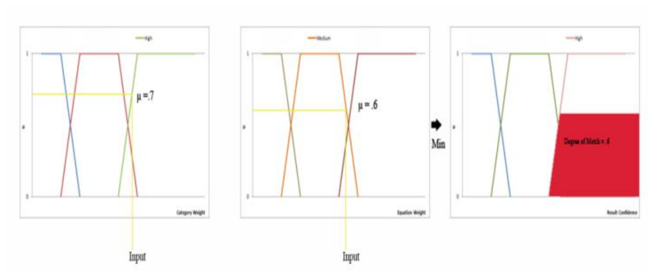


Fig. 6. Membership function of result

VI. Conclusion

A framework based on Fuzzy logic – Ontology model can be used to solve mathematical word problems. The ability to use Ontology as a tool for relational data allows this model to build equations for many types of mathematical word problems. The system is not limited to word problem types or categories described in this paper, but can be scaled to address all word problem types. The growing power of NLP gives the system this power to analyze and build equations of many kinds. The primary purpose of this model is to teach users how to solve word problems. The results returned by a search of a mathematical equation displays the steps taken to compute the solution which acts as an aid to teach users the process of solving word problems. Without displaying the proper steps that were taken to solve the problem a user would have difficulty understanding how the given answer was derived. This model contributes by aiding users in learning and helping users understand how to solve real world problems that pertain to mathematics.

References

[1] S. Tse and V. Dahl, "Earning and using mathematics software the natural way," *Applied Mathematics Letters*, vol. 15, pp. 875-879, 2002.

[2] R. Schumacher and L. Fuchs, "Does understanding relational terminology mediate effects of intervention on compare word problems?," *Journal of Experimental Child Psychology*, vol. 111, pp. 607-628, 2012.

[3] D. Cummins *et al.*, "The role of understanding in solving word problems," *Cognitive Psychology*, vol. 20, pp. 405-438, 1988.

[4] M. LeBlanc and S. Weber-Russell, "Text integration and mathematical connections: A computer model of arithmetic word problem solving," *Cognitive Science*, vol. 20, pp. 357-407, 1996.

[5] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing &amp; Management*, vol. 24, pp. 513-523, 1988.

[6] D. J. Briars and J. H. Larkin, "An integrated model of skill in solving elementary word problems," *Cognition and Instruction*, vol. 1, pp. 245-296, 1984.

[7] Y.-M. Yeh *et al.*, "Research on Reasoning and Modeling of Solving Mathematics Situation Word Problems of Primary Schools," *Multimedia Workshops, International Symposium on*, vol. 0, pp. 35-41, 2007.

[8] N. Kitani and S. Yukita, "Mathematical Ontology and a Software Tool for Semantic Retrieval of Exercise Problems," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 1654-1659, 2008.

[9] M. Yuhui *et al.*, "Frame-Based Calculus of Solving Arithmetic Multi-Step Addition and Subtraction Word Problems," *Education Technology and Computer Science, International Workshop on*, vol. 2, pp. 476-479, 2010.

[10] S. Samarasinghe and S. Hui, "Mathematical Document Retrieval for Problem Solving," *Computer Engineering and Technology, International Conference on*, vol. 1, pp. 583-587, 2009.

[11] Natural Language Toolkit. (2001). [Online]. Available: http://www.nltk.org/

[12] Q. Luo *et al.*, "A semantic term weighting scheme for text categorization," *Expert Systems with Applications*, vol. 38, pp. 12708-12716, 2011.

[13] D. L. M. Natalya and F. Noy. (2000). Ontology Development 101: A Guide to Creating YourFirst Ontology. [Online]. Available: http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

[14] N. Kitani and S. Yukita, "The educational uses of mathematical ontology and the searching tool," *Frontiers in Education, Annual*, vol. 0, 2008.

[15] L. D. A. N. Capuano, F. Orciuoli, S. Miranda, and F. Zurolo, "Ontology Extraction from Existing Educational Content to Improve Personalized e-Learning Experiences," *ICSC*, pp. 577-582, 2009.

[16] F. V. H. Deborah and L. McGuinness. (2004). OWL Web Ontology Language Overview. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-features-20040210/

Kyle F. Morton is from Macon, GA, USA. He was born on September 3, 1985.Kyle Morton is a doctoral candidate at Colorado Technical University – Colorado Springs, USA. Kyle Morton holds a Bachelors of Computer Science from Samford University, Birmingham, USA, and a Masters of Information Systems from the University of Phoenix, USA. Kyle's research interests include data mining, big data analytics, natural language processing, artificial intelligence, information systems, fuzzy system, mobile computing and security, and intelligent systems.

His current occupation is as a Software Engineer at Argo Data in Dallas, TX, USA. He has also worked with other commonly known companies such as American Airlines, Bank of America, and Hewlett-Packard. Throughout his career Kyle Morton has had the opportunity to research and develop innovative technologies which includes technologies in the areas of video streaming, banking software, and real-time software systems. He is an IEEE member.

Yanzhen Qu currently is the dean and a professor in Computer Science and Information Technology at Colorado Technical University – Southern Colorado, USA. Dr. Qu holds a B.Eng. in Electronic Engineering, a M. Eng. in Electrical Engineering, and a Ph.D. in Computer Science. Over his industrial career characterized by many "the world first innovations", he has served at various senior or executive level Product R&D and IT management positions at several multinational corporations. He was also the chief system architect and the development director of several world first very large real-time commercial software systems.

At Colorado Technical University, Dr. Qu is the dissertation supervisor of over ten computer science doctoral students, and his recent research interests include cloud computing security and architecture, cyber security risk detection and mitigation, data engineering, software engineering process and methods, soft computing, data mining over non-structured data, human-oriented computer interface, scalable enterprise information management system, big data analytics as well as embedded and mobile computing. He served as general/program/session chair or keynote speaker in various professional conferences or workshops. He has published many research papers in the peer reviewed conferences and professional journals, and is currently serving as a member of editorial board of several professional journals.