# Unit Conflict Resolution for Automatic Math Word Problem Solving

Nuwantha Dewappriya, Gimhani Uthpala Kankanamge, Dushani Wellappili,
Asela Hevapathige, Surangika Ranathunga
Department of Computer Science and Engineering
University of Moratuwa
Katubedda 10400, Sri Lanka
{nuwantha.13, gimhani.13, dushani.13, asela.13, surangika}@cse.mrt.ac.lk

## ABSTRACT
Among the statistical approaches for math word problem solving, template based approaches have shown to be more robust, against a wide spectrum of math word problems, while other approaches target simple arithmetic problems that compose of only one operation or equation. However, even template based systems are poor in performance for questions that contain different units to describe the same measurement. This paper presents a unit conflict resolution system to improve the performance and accuracy of template based systems under minimal supervision. To illustrate the importance of unit conflict resolution for math word problems, we have annotated a new dataset of 385 algebra word problems. We evaluate the performance of our approach both on a benchmark dataset and this new corpus. Experimental results show that integration of our system to an existing automatic Math word problem solver outperforms state-of-the-art results when the dataset contains different units to describe the same measurement.

## CCS Concepts
**• Applied Computing → Education → Computer-managed instruction**

**• Artificial intelligence → Natural language processing → Information Extraction**

## Keywords
math word problems; ontology; units; unit conflict resolution

## 1. INTRODUCTION
Solving math word problems is both challenging and rewarding. A math word problem consists a high level abstraction to a set of equations woven around a story. To get the right answer, we should understand the text, figure out which mathematical operations and numerical values to use, and then perform the calculations correctly. Thus, automatic math word problem solving is rather challenging to a machine.

Current trends in automatic math word problem solving could be mainly identified as statistical based approaches [1, 2, 3, 4, 5, 6], and semantic parsing approaches [7, 8]. Statistical approaches that rely on templates for ranking have been successful in addressing a wide range of question domains given the fact that training samples are available for corresponding template. Moreover, these systems are capable of solving a system of linear equations, which is a pacesetter in comparison to other attempts that focus on questions with one equation [3, 4, 5, 6].

In template based approaches by Kushman et al. [1] and Zhou et al. [2], for a given new question, the most appropriate template and assignments to its unknown values are predicted based on a trained model on ranking templates. While Kushman et al. [1]'s work has been improved in both accuracy and performance by Zhou et al.[2]'s work, this system still creates a high dimensional sparse feature space.

Exploring the templates for Kushman's ALG-514 dataset [1], it could be observed that there exist several templates that are specific versions of another. As shown in Table 1, the current template is a more specific template that includes 0.01 to demonstrate the relation between dollars and cents. As shown in this example, a unit conflict could be defined as the mismatch of units within the same problem, which adds an additional overhead of including the relationship between these units within templates. This increases the number of templates of the solver, thus affecting its performance in both accuracy and time complexity wise.

**Table 1. Template creation for a question**

| | |
|---|---|
| **Question** | A car rents for 30 dollars per day plus 23 cents per mile. You are on a daily budget of 76 dollars. What mileage can you go and stay within your budget? |
| **Current Template[i]** | Equation: 30.0+(23*0.01*miles)=76.0<br>Template: a+(b*0.01*n)=c |
| **General Template** | Equation: 30.0+(0.23*miles)=76.0<br>Template: a+(b*n)=c |

In our work[1], we focus on preventing the need of such extra templates via an ontology based unit conflict resolving approach. This ensures reduced time for feature creation, reduced time to train the model and less sparsity of the feature space. Finally, this positively affects the accuracy of the overall system.

[1] Our code is publicly available at https://github.com/Nuwantha/UnitNomalizer

We created our own dataset[2] where we have included more templates and questions having unit conflicts. These types of questions were only found in Kushman's data set [1] whereas datasets such as ARIS [3] did not include any question with this characteristic. The experiments done on both datasets proved that our solution reconstructs the corpus of questions so as to provide better prediction results even under weak supervision.

# 2. LITERATURE REVIEW

## 2.1 Automatic Math Problem Solving

Currently, statistical based approaches and symbolic semantic parsing approaches are prominent in solving Math problems.

Symbolic based approaches derive the equations from the structured representations constructed out of the problem text using semantic parsing or pattern matching. Liguda and Pfeiffer [7] have been successful in using a symbolic approach that could develop an augmented semantic network for primary level Math word questions. However, questions with the concepts of money, distances, time and volume that involve conflicting units are not handled in this research. The symbolic approach presented by Shuming Shi et al. [8] too suffers from the same limitation.

Initial statistical based approaches including the works of Hosseini et al. [3] and Roy and Roth [4] mainly focus on simple arithmetic math word problems. The system of Hosseini et al. [3] is based on the task of learning to solve word problems by mapping the verbs in the problem text into categories that describe their impact on the world state. Since this approach works only with problems with one unit, more domain knowledge is needed to deal with the missing information in handling problems with multiple unit conflicts. Roy and Roth [4],[5],[6] used a graph based statistical approach to express the relationship between quantities and units as trees to generate arithmetic equations. This approach too is not capable of addressing problems with conflicting units and percentages. All these approaches blindly extract numerical values from the question text which causes negligence over unit conflicts.

Leszczynski and Moreira [9] have used a recurrent neural network to solve physics word problems on gravitational acceleration. Their grammar allows mixed units (SI and US customary). However, the domain is restricted to the scope of physics problems, and such deep learning approaches have the difficulty of collecting a large data set to scale for common arithmetic word problems.

Template based statistical approaches have the advantage of the integrated structure of its equation templates to formulate all kinds of mathematical concepts observed in the training data. Kushman et al. [1] have succeeded in using a template-based approach to solve complex problems using a set of linear equations. The problem set of Kushman et al. [1] is complex because it consists of questions with more than one variable while covering a wide variety of problem domains including interest, ticket purchasing, ratio, height comparing, finance and physics, etc. The mapping of natural language into equation templates is done by examining the features of the training problems. Zhou et al. [2] have used quadratic programming to increase the efficiency and accuracy on Kushman et al.'s [1] work. They have considered only the numerical values in questions and their associated features in place of considering both numerical values and nouns related to each numerical value. However this still has the drawback of needing more hypothesis space to generate the answer, because there exist many possible assignments to each slot of the template. Further, this approach needs to add more templates to the template set whenever a new question comes from a different domain with mixed measurement units. This increase of number of templates has a dramatic effect to the overall accuracy. In general, there are problems that can be solved by the same set of equation templates. However, the two problems will only differ from the fact that one contains whole numbers and other contains as percentages. Upadhyay et al. [11] have followed the same approach as Kushman et al. [1] and introduced a semi automatically annotated dataset for the evaluation of automatic solvers.

None of these existing template based approaches has not integrated background knowledge to extract the minimal template after resolving the conflicts of mixed units.

## 2.2 Ontologies of Unit Measurements

Unit conflict resolution has been an important topic in different domains including bioinformatics and other exact sciences [15]. There are several choices developed to address this formalization of unit representation and integration. Such notable ontologies of are Measurement Unit Ontology (MUO) [12], Ontology for Engineering Mathematics (EngMath) [13], Quantities, Units, Dimensions and Types (QUDT) [14], and the Ontology of Unit of Measure (OM) [15]. The framework of MUO [12] supports automated conversion of SI units, but it has the limitation of formula based conversion between SI units and other similar units in other unit systems. EngMath [13] Ontology does not have such issue but has the limitation of not available in any language other than the written language, Ontolingua [16]. The high level design features of OM [15] and QUDT [14] are similar, but OM [15] outperforms the other by defining OWL classes, which facilitates logical reasoning.

# 3. UNIT CONFLICT RESOLUTION

Unit conflict resolution includes converting the problem into a normalized stage where all the numerical terms are in compatible and required units.

In math word problems (elementary and secondary), the units used are usually of different systems of units and of different scales. Therefore, in order to resolve conflicts, we can mainly identify two types of conversions between them.

## 3.1 Conversion between Same System of Units

This category includes questions from the same system (from SI unit system, British unit system etc). As shown in Table 2, when a question includes both month and year whereas the answer is expected in months, a unit conversion is done so that all the units are converted to months before fed into the parser. This will prevent the need of extra template that contains '*12' to exploit the domain knowledge.

**Table 2. Conversion between same system of units**

| | |
|---|---|
| **Original Question** | The last accurate measurements of the Jakobshavn Glacier in Greenland have it travelling at **5.25 kilometers** in a **5 month** period. At this rate, how far does it travel in **one year**? |
| | Equation:<br>(5.0*m)+-5.25 = 0, (-1.0*12.0*m)+n = 0<br>Template: |

| | |
|---|---|
| | (a*m)+-b = 0, (-c*12.0*m)+n = 0 |
| **Conflicting Units** | year = 12.0* months |
| **Normalized Question** | The last accurate measurements of the Jakobshavn Glacier in Greenland have it travelling at **5.25 kilometers** in a **5 month** period. At this rate, how far does it travel in **12 months**? |
| | Equation:<br>(5.0*m)+-5.25 = 0, (-12.0*m)+n = 0<br>Template:<br>(a*m)+-b = 0, (-c*m)+n = 0 |

## 3.2 Conversion between Different Systems of Units

This category includes the questions that consist of measures from several unit systems. For example as shown in the Table 3, if the same problem includes feet and meters, a unit conversion is done so as to convert meters to feet. While Kushman's approach [1] embedded the detail on the relation in template, we suggest reconstructing the question to include such information.

**Table 3. Conversion between different systems of units**

| | |
|---|---|
| **Original Question** | **Two meters** of fabric will cost **100 dollars**, how much will **2 feet** of fabric will cost? |
| | Equation:<br>(2.0*3.28*m)+-100.0 = 0, (-2.0*m)+n = 0<br>Template:<br>(a*3.28*m)+-b = 0, (-c*m)+n = 0 |
| **Conflicting Units** | meters= 3.28 * feet |
| **Normalized Question** | **6.56 feet** of fabric will cost **100 dollars**, how much will **2 feet** of fabric will cost? |
| | Equation:<br>(6.56*m)+-100.0 = 0, (-2.0*m)+n = 0<br>Template:<br>(a*m)+-b = 0, (-c*m)+n = 0 |

# 4. METHODOLOGY

In this paper, we present an ontology-based unit conflict resolution system to resolve unit conflicts in math word problems. This acts as a preprocessor to the math word problem solver.

Based on the units and conversions required, an ontology based approach was more promising than a rule based or statistical based system for unit conversion process. A rule based system suffers from the fact that a vast scope of rules need to be analyzed to complete the conversion while a statistical based system is not guaranteed to give the accurate answer in such context. Since this converter acts as a plugin amidst the process, accuracy of conversion being 100% and no dramatic reconstruction of the question should be guaranteed.

As specified in the related work, Ontology of Measure [15] is the most prominent ontology that covers a wide range of units while supporting the necessities for our system.Our ontology-based unit conflict resolution system composes a pipeline that consists of a tokenizer, numerical token identifier, unit frame recognizer, unit conflict identifier and unit conflict resolver (normalizer) as shown in the Figure 1.

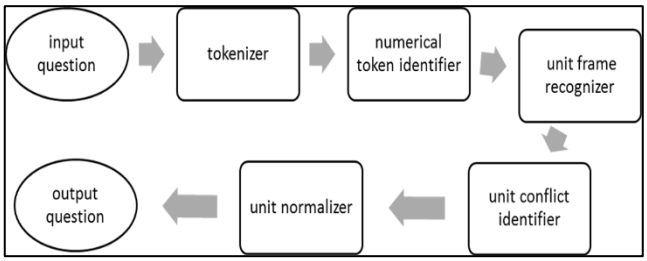


**Figure 1. Unit conflict resolution system pipeline**

Each question is traced to identify numerical tokens, which are then embedded in unit frames. Unit frames carry information related to each value through the pipeline. Unit conflict identification process makes use of related unit, its base unit and the unit represented in question sentence as shown in the Figure 2. All the unit frames relevant to a question are examined by the system to detect if the question needs unit conflict resolving.

Unit Frame
-unit
-base unit
-value
-is question unit

**Figure 2. Unit frame**

The base unit of each unit is extracted via the ontology. By examining the unit frame set of a question, we detect whether there exist any unit frames that belong to same unit category but different unit scales. Such unit frame subsets are evaluated and updated. The relationship between base unit and the current unit are extracted through the ontology. The simplified pseudocode for this process could be demonstrated as in Figure 3.

```
For each question:
 -For each numeric value:
  -Create unit frame

 -For each unit frame in question:
  -If unit conflict exists
   -If base units equal unit:
    -If unit_frame exists in question sentence:
     -Convert each unit in unit_frame set to
question_unit
    -Else:
     -Convert each unit to base unit
 -Update question with new units
```

**Figure 3. Pseudocode for unit conflict resolution**

At the end of this pipeline, the input question set is updated so that it is ready to be added to the common flow of the current template based approaches.

# 5. EXPERIMENTS

## 5.1 Experimental Setup

We used the quadratic programming approach by Zhou et al. [2], which is an improved version of Kushman et al. [1]'s work as our baseline.

In order to compare and contrast the effect of unit conflict resolution, we created our own data set. Our dataset includes 385 questions, and it complies with the conditions (e.g. minimum number of questions per template) used by Kushman et al. [1] in creating the dataset so as to be consistent in training. This dataset is based on web resources, variations on questions from other datasets and newly created questions. The data set includes 73 questions that have unit conflicts within the question. These questions with unit conflicts belong to 11 different templates (Kushman et al.[1]'s data set includes 79 questions with unit conflicts for 3 templates). In both of these data sets, a more generic version of equations is included to replace all of these templates with unit conflicts.

## 5.2 Experimental Results

Template based approaches by Kushman et al.[1] and Zhou et al.[2] use two different levels of supervision as semi-supervised (weakly supervised) and fully supervised. While the fully supervision provides labeled equations for all the questions, semi-supervision provides labeled equations for a subset only. We have tested our unit conflict resolution system against both supervision levels with 5-fold cross validation.

Table 4 shows the results on our dataset. A significant reduction in the number of templates thus affecting number of features and training time could be easily identified from this result. This has been able to increase the accuracy of supervised system upto 74.2% from 67.5% under fully supervision. Semi-supervision has led the accuracy upto 76.38% from 69.92%. This provides the fact that after questions are normalized, a greater accuracy could be gained with less supervision.

The reduction in time for training model too is reduced by approximately 40%. The template reduction could be explained as follows. Normalizing ensures that templates are free from domain knowledge embedded to them. Table 5 provides examples of template substitutions occurred in our data set.

**Table 4. Experimental results on our dataset**

| | **Original approach** | | **After unit conflict resolution** | |
|---|---|---|---|---|
| | **Supervised** | **Semi-supervised** | **Supervised** | **Semi-supervised** |
| **#templates** | 22 | 22 | 15 | 15 |
| **#features** | 103936 | 98531 | 66235 | 62247 |
| **Training time** | 620s | 631s | 433s | 380s |
| **#correct predictions** | 260 | 249 | 286 | 272 |
| **#incorrect predictions** | 125 | 107 | 99 | 84 |
| **Accuracy** | 67.5325% | 69.922% | 74.2858% | 76.385% |

**Table 5. Template reduction in our dataset**

| **Original template** | **Substituted template** |
|---|---|
| (100.0+a)*0.01*m+-b = 0 | (1.0+a)*m+-b = 0 |
| (a*3.0*m)+-b = 0, (-c*m)+n =0 | (a*m)+-b = 0, (-c*m)+n=0 |
| (a*m)+-b = 0, (-c*24.0*m)+n = 0 | (a*m)+-b = 0, (-c*m)+n = 0 |
| (a*m)+-b = 0, (-c*12.0*m)+n = 0 | (a*m)+-b = 0, (-c*m)+n = 0 |
| (a*0.01*b*0.01*c)+-m = 0 | (a*b*c)+-m = 0 |
| (a*m)+-b =0,(-c*60.0*m)+n =0 | (a*m)+-b = 0, (-c*m)+n = 0 |
| (a*m)+(-b/60.0)=0,(-c*m)+n= 0 | (a*m)+-b = 0,(-c*m)+n = 0 |

Results on Kushman et al.[1]'s data set are shown in Table 6. Kushman's data set includes 514 questions out of which 79 questions are resolved for unit conflict resolution by our system. Since these questions belong to 3 templates only, we cannot observe a positive trend in accuracy under fully supervision. However, semi-supervision has shown a rise in accuracy. This is quite significant because under weak supervision, we could get better results with unit conflict resolution. Table 7 presents the original templates and templates after unit conflict resolution. These substituted templates are already available as templates for other questions.

Moreover, this makes feature space reduction by 25,000 in both instances. The sparsity of feature space is a significant challenge in template based approaches. This becomes worse when more questions belonging to different templates are added. Thus, these experimental results exhibit the importance of unit conflict resolution to handle sparsity of feature space. Further, a significant reduction in time to train the model too is observed.

**Table 6. Experimental results on Kushman's dataset [1]**

| | **Original approach** | | **After unit conflict resolution** | |
|---|---|---|---|---|
| | **Supervised** | **Semi-supervised** | **Supervised** | **Semi-supervised** |
| **#templates** | 26 | 26 | 23 | 23 |
| **#features** | 219346 | 219346 | 190255 | 190253 |
| **Training time** | 955s | 1811s | 877s | 1534s |
| **#correct predictions** | 411 | 378 | 408 | 384 |
| **#incorrect predictions** | 103 | 136 | 106 | 130 |
| **Accuracy** | 79.9297% | 73.3109% | 79.3341% | 74.6726% |

**Table 7. Template reduction in Kushman's dataset**

| Original template | Substituted template |
|---|---|
| (a*0.01*m)+(b*0.01*n)+-c = 0, -d+m+n = 0 | (a*m)+(b*n)+-c = 0, -d+m+n = 0 |
| a+-b+(-c*0.01*m) = 0 | a+-b+(-c*m) = 0 |
| (a*0.01*m)+(b*0.01*n)+(-c*d*0.01) = 0, -d+m+n = 0 | "(a*m)+(b*n)+(-c*d) = 0, -d+m+n = 0 |

Overall, these experiments show that the use of unit conflict resolution is a positive step towards gaining high accuracy (via template reduction and feature space reduction) for template based math word problem solving systems. Further, unit conflict resolution gives promising results even with weak supervision due to the fact that questions are in a normalized state. This could be of huge advantage to future research on automatic math word problem solving to identify the importance of question normalization to reduce diversity and complexity.

# 6. CONCLUSION

Automatic math word problem solving is still a challenging problem. A wide variety of math word problems is the most prominent reason behind this. Preprocessing that could reformat questions so that they are less diverse is an important step that should be given attention in automatic math word problem solving. This assures that problems could be made more similar so that a clean corpus could be made ready for training. In this paper, we emphasized this in relation to solving unit conflicts. Our focus on future work is to investigate the possibility of using statistical systems to get perfect unit conflict resolution.

# 7. ACKNOWLEDGEMENTS

We thank Zhou Lipu on the guidance given to us in understanding the current research work on automatic math word problem solving.

# 8. REFERENCES

[1] N. Kushman, L. Zettlemoyer, R. Barzilay and Y. Artzi, "Learning to Automatically Solve Algebra Word Problems," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 523-533, 2014.

[2] L. Zhou, S. Dai and L. Chen, "Learn to Solve Algebra Word Problems Using Quadratic Programming," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 817-822, 2015.

[3] M. J. Hosseini, H. Hajishirzi, O. Etzioni and N. Kushman, "Learning to Solve Arithmetic Word Problems with Verb Categorization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 523-533, 2014.

[4] S. Roy and D. Roth, "Solving General Arithmetic Word Problems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[5] S. Roy and D. Roth, "Illinois Math Solver: Math Reasoning on the Web," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 52-56, 2016.

[6] S. Roy and D. Roth, "Unit Dependency Graph and its Application to Arithmetic Word Problem Solving," in *Association for the Advancement of Artificial Intelligence*, 2017.

[7] C. Liguda and T. Pfeiffer, "A Question Answer System for Math Word Problems," in *First International Workshop on Algorithmic Intelligence*, 2011.

[8] S. Shi, Y. Wang, C.-Y. Lin, X. Liu and Y. Rui, "Automatically Solving Number Word Problems by Semantic Parsing and Reasoning," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1132-1142, 2015.

[9] D. G. Bobrow, "A question answering system for high school algebra word problems," in *Proceeding AFIPS '64 (Fall, part I) Proceedings of the October 27-29, 1964, fall joint computer conference, part I*, 1964.

[10] M. Leszczynski and J. Moreira, "Machine Solver for Physics Word Problems," in *30th Conference on Neural Information Processing Systems (NIPS 2016)*, pp. 2-5, 2016.

[11] S. Upadhyay and M.-W. Chang, "Annotating Derivations: A New Evaluation Strategy and Dataset for Algebra Word Problems," 2016.

[12] "MUO : Measurement Units Ontology," [Online]. Available: http://idi.fundacionctic.org/muo/muo-vocab.html. [Accessed 5 July 2017].

[13] T. R. Gruber and G. R. Olsen, "An Ontology for Engineering Mathematics," in *Proceedings of Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany, 1994.

[14] QUDT.org, "QUDT - Quantities, Units, Dimensions and Types," [Online]. Available: http://www.qudt.org/. [Accessed 6 July 2017].

[15] H. Rijgersberg, M. Van Assem, and J. Top, "Ontology of units of measure and related concepts," *Semant. Web*, vol. 4, no. 1, pp. 3–13, 2013.

[16] "Ontolingua Home Page," [Online]. Available: http://www.ksl.stanford.edu/software/ontolingua/. [Accessed 5 July 2017].

# Authors' background

| Your Name | Title* | Research Field | Personal website |
|---|---|---|---|
| Nuwantha Dewappriya | Undergraduate | Natural Language Processing | |
| Dushani Wellappili | Undergraduate | Natural Language Processing | |
| Gimhani Uthpala Kankanamge | Undergraduate | Natural Language Processing | |
| Asela Hevapathige | Undergraduate | Natural Language Processing | |
| Dr. Surangika Ranathunga | Senior Lecturer | Natural Language Processing | |

***This form helps us to understand your paper better, the form itself will not be published.**

***Title can be chosen from: master student, Phd candidate, assistant professor, lecturer, senior lecture, associate professor, full professor**