Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3479–3490
Marseille, 20-25 June 2022

HAWP: a Dataset for Hindi Arithmetic Word Problem Solving

Harshita Sharma, Pruthwik Mishra, Dipti Misra Sharma
LTRC, IIIT-Hyderabad
{harshita.sharma, pruthwik.mishra}@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

Word Problem Solving remains a challenging and interesting task in NLP. A lot of research has been carried out to solve different genres of word problems with various complexity levels in recent years. However, most of the publicly available datasets and work has been carried out for English. Recently there has been a surge in this area of word problem solving in Chinese with the creation of large benchmark datastes. Apart from these two languages, labeled benchmark datasets for low resource languages are very scarce. This is the first attempt to address this issue for any Indian Language, especially Hindi. In this paper, we present HAWP (Hindi Arithmetic Word Problems), a dataset consisting of 2336 arithmetic word problems in Hindi. We also developed baseline systems for solving these word problems. We also propose a new evaluation technique for word problem solvers taking equation equivalence into account.

Keywords: word problem solving, lexical diversity, low resource, biLSTM

1. Introduction

Math Word Problems (MWPs), mainly taught in primary and high schools, are considered to be an integral part of Math curriculum across the globe. MWPs are seen as brain teasers because they expect the solver to understand a hypothetical situation involving quantities using some statements that must be used to answer a question about that situation. For a machine to correctly model this type of information, form relationships among the quantities and generate the answer is a very complex task. Most often MWPs are designed as real world narratives and act as a mental stimuli for the students. Recent works have shown that solving elementary-level word problems itself still poses a significant challenge for the NLP community. Although difficult, word problem solving has been approached with a number of innovative and intelligent perspectives: semantic parsing, template matching, explainable solvers etc.

As a result of these different approaches, a large number of datasets of various levels of complexity have been developed in languages like Chinese and English. However, no substantial dataset exists for Indian Languages. To address this, we have created a dataset for Hindi Word Problem Solving. To make the data more diverse we adopted a 2-pronged strategy to construct a diverse and challenging dataset of 2336 MWPs. Some of these MWPs were collected from Hindi textbooks and worksheets and also from Hindi-medium educators while most of them were augmented using translation. Through our paper, we make the following contributions:

- We tackled the problem of lack of Word problem solving dataset and solvers in Hindi language.
- We designed a good quality, diverse and challenging, publicly available[1] dataset of 2336 MWPs annotated with equations, number of operations and indices of relevant quantities in the word problems.
- We crafted guidelines that can help augment more Hindi MWPs using translation.
- We propose baseline systems and equation equivalence technique to handle multiple possibilities for equations.

2. Related Work

Several large scale datasets have been released over the years for Mathematical word problem solving like AQuA (Ling et al., 2017) containing 100K complex MWPs and MathQA (Amini et al., 2019) containing 37K word problems in English and Chinese dataset Ape210K (Zhao et al., 2020) containing 210K problems and 56K templates. Recently, the focus has shifted from large sized datasets to more diverse datasets. (Miao et al., 2020) have pointed out the challenges of skewed lexical diversity, difficulty level and problem type distribution of MWPs, incorrectly annotated equations and answers in large MWP datasets. (Patel et al., 2021) introduced a challenge dataset SVAMP for which the best accuracy of state-of-the-art solvers is much lower. (Roy and Roth, 2018) also showed that the benchmark datasets are biased and include word problem with high lexical overlapping.

Arithmetic word problem solving has always remained a challenge for the NLP community from the 1960s (Bobrow, 1964) since it involves natural language understanding, identification of relevant and irrelevant quantities, operations as well semantic reasoning across sentences. Initial solvers were rule based, schema based capable of solving only a few word problems with very limited vocabulary coverage. Next came the statistical solvers which tried to learn the alignment of variables and numbers in the equation

[1] `https://github.com/hellomasaya/hawp`

templates. KAZB (Kushman et al., 2014) was the first attempt to learn these alignments for a set of linear equations. Other statistical approaches (Hosseini et al., 2014) used verb categorization for solving word problems where verbs triggered the flow of quantities between entity driven containers. (Roy and Roth, 2016) used expression trees to solve word problems where the whole solving process is decomposed into multiple classification tasks involving quantities and operations. All these methods relied on lexical, structural, dependency, wordnet, other manually crafted features for the classification. The major drawback of statistical systems is their inability to perform well on larger and diverse datasets. A simple similarity based retrieval model (Huang et al., 2016) outperformed its sophisticated statistical counterparts on large datasets. (Wang et al., 2017) was the first one to reduce the problem into a sequence to sequence learning problem. Although many neural approaches have reported state-of-the-art performance on benchmark datasets, (Patel et al., 2021) show that most of the current solvers are not robust and minor changes in the input word problem can degrade the performance. Most of these efforts are centred either around English or Chinese. There have been very few attempts to develop word problem solvers for other low resource languages. India being a country where multiple languages are spoken, the need for creating word problem datasets and developing efficient solvers is paramount. To the best of our knowledge, this is the first dataset for arithmetic word problems in any Indian language.

3. HAWP

In this section, we introduce a new dataset in Hindi, HAWP (Hindi Arithmetic Word Problems) for the task of Arithmetic Word Problem Solving in Hindi. HAWP is a collection of 2336 MWPs dealing with addition, subtraction, multiplication and division operations having one unknown. The dataset covers one and two operation word problems.

3.1. Dataset Construction

Building a rich dataset for a low resource language like Hindi is a complex task, specially since no MWP repository is already available. Though MWPs are a crucial part of Math curriculum in Hindi-medium schools and examinations, it is not easy to find diverse data. That being the case, we constructed this dataset using a number of methods:

- **Manually Crafted Problems:**
  - Hindi-medium Math Teachers: We asked Math teachers from Hindi-medium schools to come up with some word problems of 1 to 6 grade level as a worksheet/exam paper they would make for their students.
  - Hindi-medium Math Textbooks: We went through many publicly available Hindi-medium Math textbooks and workbooks for grade 1 to 6 and constructed Hindi word problems similar to the textbooks so that the naturalness of the word problems can be maintained.
- **Data Augmentation using Translation:** Around 1600 problems from different benchmark datasets in English, namely AI2 (Hosseini et al., 2014), Unbiased (Roy and Roth, 2018), ASDiv (Miao et al., 2020) were translated for augmenting MWPs in Hindi. Section 3 provides an in-depth description of how problems from these English datasets were translated as a part of data augmentation.

Method	#MWPs
Manually Crafted	736
Augmentation	1600
Total	2336

Table 1: Various methods of dataset construction

3.2. Annotation

The crafted and the augmented data was annotated with equations, number of operations and indices of relevant quantities in the word problems. We also developed an equation annotation tool to facilitate easy annotation of equations and the relevant indices.

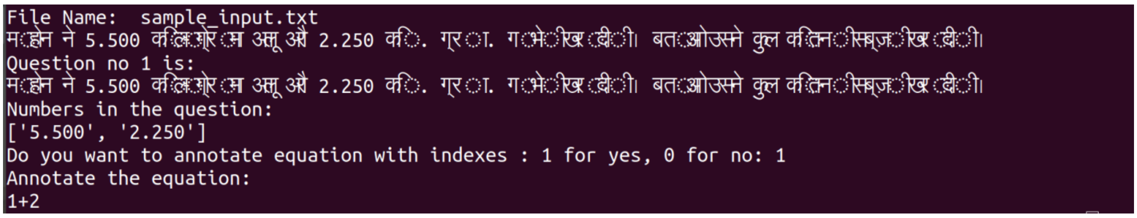
Operation	#MWPs
1-operation	1786
2-operation	550
Total	2336

Table 2: Problems with different operations

3.2.1. Equation Annotation Tool

The annotation of equation and relevant quantities was done in house by the authors. In order make this manual effort less tiresome and error free, we developed a simple command line based annotation tool. The tool first displays the question, lets the user type the equation and other relevant information and saves it to a file. The user just needs to identify the quantities and operations in the equation. This was done in order to avoid entering large numbers or fractions which could become a major source of errors. The annotation tool also had the facility to convert a number written in a word form into its numerical equivalent e.g. three gets converted to 3. The information annotated by the tool for the example shown in figure 1 is as follows:

- Question: *mohan ne 5.500 kilograam aaloo aur 2.250 ki. gra. gobhee khareedee. batao usane kul kitanee sabzee khareedee.*
- English Gloss: *Mohan bought 5.500 kilograms of potatoes and 2.250 kg of cauliflower. Find the total weight of the vegetables that he bought.*



```
File Name:  sample_input.txt
मोहन ने 5.500 किलोग्राम आलू और 2.250 कि. ग्रा. गोभी खरीदी। बताओ उसने कुल कितनी सब्ज़ी खरीदी।
Question no 1 is:
मोहन ने 5.500 किलोग्राम आलू और 2.250 कि. ग्रा. गोभी खरीदी। बताओ उसने कुल कितनी सब्ज़ी खरीदी।
Numbers in the question:
['5.500', '2.250']
Do you want to annotate equation with indexes : 1 for yes, 0 for no: 1
Annotate the equation:
1+2
```

Figure 1: Snapshot of the Equation Annotation Tool

- Equation: $X = (5.550 + 2.250)$
- Relevant Indices: 0, 1 where 0 refers to the first quantity and 1 refers to the second quantity in the problem text

3.2.2. Inter-Annotator Agreement

Two annotators were involved in the task who had prior experience in automatic word problem solving. They annotated single variable equations for 100 word problems. If both the equations match or are equivalent, we consider it an agreement and disagreement otherwise. We found 94% agreement among the annotators. The disagreements can be categorized into 2 types. The first kind of disagreement occurred due to the incorrect identification of operands or operations. The other kind was due to different conversions of units in unit problems. An example of type 2 disagreement

- Question: *reena baajaar se 1.400 kigra tamaatar tatha 750 graam mirch khareed kar apane thaile mein rakhatee hai. usake thaile ka kul bhaar kitana hai ?*
- Equation 1: $X = (1.400 + (750/1000))$ in kigra or kg
- Equaiton 2: $X = (1400 + 750)$ in grams

In this example, the disagreement is due to the choices of unit conversion. Overall frequencies of Type 1, Type 2 disagreements were 4 and 2 respectively.

4. Augmentation of Hindi Word Problems

The number of naturally Hindi word problems acquired from textbooks and teachers were significantly low for a NLP dataset. To solve this problem, we augmented the data by translating word problems from English datasets - AI2 (Hosseini et al., 2014), Unbiased (Roy and Roth, 2018), ASDiv (Miao et al., 2020). This augmentation task is being motivated by the fact that the corresponding English datasets hold various kinds of problems: different number of unknowns, irrelevant information in problems, problems requiring world-knowledge etc. leading to a richer and diverse Hindi dataset as well. Moreover, all three datasets are different from each other in terms of length of each word problem and its complexity.

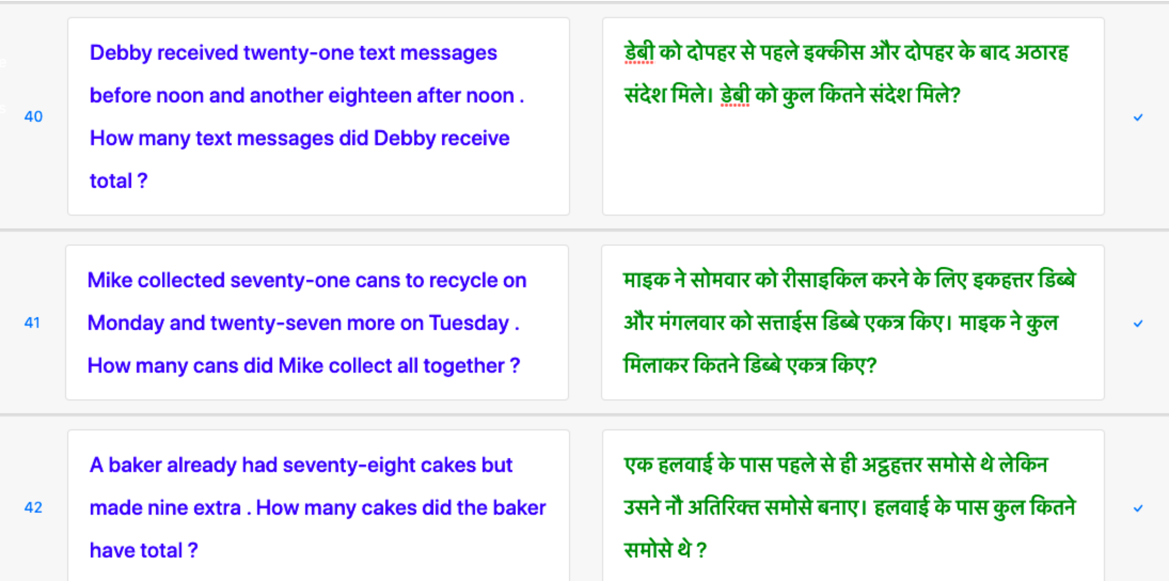
The task of translation was performed by professional translators in two steps. A batch of randomly selected English word problems were manually translated to Hindi. To minimise time and effort, we decided to translate the rest of the problems using machine translation tools. Before doing so, we performed machine translation (MT) on the same batch of word problems that were manually translated and compared the two translations to find issues in the machine-translated word problems. To find and correct these issues a post-edit tool was used where translations from multiple MT tools like Google Translate and in-house systems were provided for reference. Correction of these issues along with other challenges are explained in the following subsections.

4.1. Challenges in data augmentation

- **Errors encountered in data augmentation using Machine Translation** The raw Hindi translations obtained from MT had a lot of errors. These errors were grouped together under various categories and each type of error was handled following a defined procedure. These errors have been documented along with their % frequency in the dataset in Table 3.
- **Missing one-to-one mapping.** Even though we embodied localisation and globalisation in our dataset, there were yet instances where we faced problems with some foreign concepts because one-to-one mapping doesn't exist for all English-Hindi words/concepts. For example:
  - The two different English concepts, 'running' and 'sprinting' get translated to the same 'daudna' in Hindi.
  - On the other hand the same verb 'serve' are translated to 'parosna', 'daalna', 'dena' depending on the recipient.
- **Cultural differences.** Since we took benchmark datasets in English, we noticed some cultural differences in word problems in the kind of objects, events, names used; which made the translated Hindi word problems less natural.

4.2. Guidelines Followed while Augmenting MWPs in Hindi

To handle the problems in the augmented data and to make sure that the dataset does not stray away from the typical nature word problems follow and at the same



40	Debby received twenty-one text messages before noon and another eighteen after noon . How many text messages did Debby receive total ?	डेबी को दोपहर से पहले इक्कीस और दोपहर के बाद अठारह संदेश मिले। डेबी को कुल कितने संदेश मिले?	✓
41	Mike collected seventy-one cans to recycle on Monday and twenty-seven more on Tuesday . How many cans did Mike collect all together ?	माइक ने सोमवार को रीसाइकिल करने के लिए इकहत्तर डिब्बे और मंगलवार को सत्ताईस डिब्बे एकत्र किए। माइक ने कुल मिलाकर कितने डिब्बे एकत्र किए?	✓
42	A baker already had seventy-eight cakes but made nine extra . How many cakes did the baker have total ?	एक हलवाई के पास पहले से ही अट्ठहत्तर समोसे थे लेकिन उसने नौ अतिरिक्त समोसे बनाए। हलवाई के पास कुल कितने समोसे थे ?	✓

Figure 2: Snapshot of the Post Editing Tool.

Error Category	Sub-category	%Frequency in machine translated word problems
Syntactic & Grammatical issues	Incorrect TAM, PP-attachment, missing pluralisation, incorrect postpositions or case marking, incorrect named entity span	34.27
Semantic issues	Wrong sense of words, Incorrect translation of phrasal verbs and participles	29.52
Discourse issues	Inconsistent honorifics, Inconsistent translation of the same word	9.42
Others	Literal translation of borrowed words, Transliteration of known concepts, Missing or Extra translation, Wrong Numeral Translated	5.24

Table 3: Types of errors encountered while augmenting Hindi word problems using Machine Translation. Examples in Table14 (Appendix)

time has Hindi fluency and grammatical correctness, we laid down some guidelines for both human translation as well as post editing. These guidelines were circulated to the post-editors with sufficient examples. These can be used as a standard guideline for future MWP data augmentation for ILs.

4.2.1. Localisation

For the translated MWPs to correctly adapt to India or more specifically to the Hindi language, we used localisation. This process was carried out by a group of native Hindi speakers. Here are some of the most frequent localisation changes applied to the translations to include the local customs and habits in the dataset:

- Foreign names like Ronald, Tiffany etc. were changed to Indian names like Madhav, Beena etc.

- Foreign currency was changed to Indian currency in all instances except when the context required foreign currency.

- Imperial and U.S. Units of measurement were changes to SI or Indian equivalents.

- Food items, sports' names, festival names etc. were changed to their Indian counterparts or similar concepts that exist in Hindi. Some examples are shown in Table 4.

Group	Source (English)	Translation (Hindi)
Currency	dollars, pennies, quarters, dimes, bill	rupay, paise, note
Units of Measurement	pounds, ounces, gallons, mile	kilogram, litre, kilometer, meel
Food items	candy, Skittles, M&Ms, pie, cookies, noodles	toffee, mithai, jalebi, biskut, maggi
Sports & Festivals	baseball, Halloween, Thanksgiving	cricket, Diwali, Holi

Table 4: Examples of Localisation

4.2.2. Borrowing

While we most certainly paid attention to localise the dataset, we did not shy away from transliterating some words and parts of word problems for which the corresponding concepts have been borrowed in India, especially in the Hindi language as long as they didn't hinder the naturalness of the sentence. Some examples can be found in Table 5

Group	Words
Food items	pizza, cake, chocolate, pastry, pasta, soup, chicken wings
Sports & Games	basketball, football, match, video games, racing game, batman game
Others	card, can, star, mixture, company, mall

Table 5: Examples of Borrowing

4.2.3. Naturalness

While translating word problems, we focused on making the translated Hindi word problem as natural as possible instead of sticking to the English counterpart. Moreover, beside making the word problems natural linguistically, we tried to make them more natural in their nature as word problems. This has been demonstrated with the help of the following word problem which is natural in Hindi:

> bina ke paas 63 mithaiyaan hain. mere paas 50 mithaiyaan hain. hamaare paas kitanee mithaiyaan hain?

However, if "kul" is added to the question, the problem becomes more natural:

> bina ke paas 63 mithaiyaan hain. mere paas 50 mithaiyaan hain. hamaare paas kul kitanee mithaiyaan hain?

This shows how typically a word problem is crafted in Hindi. So we tried to bring in this property as well. Some examples can be found in Table 12 (Appendix).

4.2.4. Grammatical Correctness

Some grammatical mistakes were found not only in the machine translated word problems but in the source (English data) as well. The reason behind this can be linked to these datasets being created using crowdsourcing. The identified mistakes were also corrected as part of post editing. Table 12 (Appendix) shows some examples and how and why they were corrected.

4.2.5. Diversity of MWPs

When it comes to diversity of a dataset, the more the better. The benchmark datasets used for augmentation have different types of word problems. Unbiased dataset (Roy and Roth, 2018) has a good number of MWPs which have introduced irrelevant information while ASDiv (Miao et al., 2020) has MWPs with very high lexical diversity.

Other than presence of irrelevant information and lexical diversity, researches have shown that there are other ways to increase the quality of a dataset. (Patel et al., 2021) have stressed on the importance of having challenging problems that pose a real test on the attention and reasoning ability of solvers. A minute change in the word problem can change its answer. Therefore, adding or changing a small part of the word problem is capable of changing the word problem itself, creating a new variant. To get the correct answer of these variants of the same problem, the solver is required to pay attention to even the smallest change in the word problem as well as to the question.

Moreover, while going through the publicly available Hindi-medium Math textbooks, we noticed not all MWPs are explicit in what they are stating and asking and it is left to the reasoning ability of the solver to understand that information. Two of the most common evidences of this are requirement of world knowledge like unit conversion, week-day, month-day conversions etc. and heavy use of ellipsis. In the example mentioned below, the implicit version of the word problem shows the possibility that the 'kharagosh' (rabbits) might have eaten some other 'aaloo' (potatoes).

Implicit: faatima ke bageeche mein 8 aaloo the. kharagoshon ne 3 kha lie. faatima ke paas ab kitane aaloo hain?

Explicit: faatima ke bageeche mein 8 aaloo the. kharagoshon ne un aalooon mein se 3 kha lie. faatima ke paas ab kitane aaloo hain?

Therefore, during augmentation we included variants of the same problem by changing the question such that it targets a different part of the problem or by changing some parts of the problem which may change the degree of explicitness, structure or information of the statements as shown in Table 13 (Appendix). These changes may or may not change the answer to the word problem.

5. Evaluation of Dataset

The developed dataset was evaluated on two parameters:

- How natural are the problems for school children for solving
- The lexical diversity of the dataset

5.1. Solvability of MWPs by Students Enrolled in Hindi-Medium Schools

Given the primary users of MWPs are students, the comprehensibility and solvability of problems in a dataset by the students studying in the target language and grade level is of utmost importance to map not only the dataset, but also the problem as close as possible to the real-world data and problem respectively. On that account, we asked Hindi-medium school students to solve some of our translated word problems. To ensure their weakness in Mathematics does not interfere with their ability to comprehend and solve these problems, we gave these problems to students of Grade 6-7. Problems were picked randomly and grouped into batches of 15. Each student was asked to solve one batch of problems.

Students were able to form correct equations for 90% of the MWPs, out of which 88.33% of them were solved with correct answers. Almost 85% of the incorrectly solved problems required 1-operation calculation. These scores show that HAWP has natural MWPs that are closer to real-world data as seen by students in their academic life. This evaluation was seen as a test for the extent of localisation, borrowing and naturalness of the dataset, and HAWP cleared this challenge.

5.2. Diversity of MWPs in the Dataset

To measure the degree of diversity of problems in HAWP, we used a number of diversity metrics proposed by different researches. For metrics dealing with lexical diversity, we understand that different languages can have different lexical overlap, hence the scores for English datasets have been listed only for reference.

5.2.1. MAWPS Lexical Diversity

We calculated the lexical overlap of HAWP as proposed by (Koncel-Kedziorski et al., 2016) which find the mean of the Jaccard Similarity for unique unigrams and bigrams of all pairs of problems. Hence, the lexical overlap of a dataset D has been formally defined as:

$$Lex(D) = \frac{1}{N} \sum_{\substack{p_i, p_j \in D \\ i<j}} PairLex(p_i, p_j)$$

where

$$PairLex(p_i, p_j) = \frac{|W(p) \cap W(q)|}{|W(p) \cup W(q)|}$$

and $W(p)$ denotes the set of unique unigrams and bigrams in a problem p and N is the number of problem pairs in D i.e. $\binom{|D|}{2}$. This metric ranges from 0 to 1 and a lower value indicates the corpus is more diverse. Lexical Diversity for MAWPS for single equation problems is found to be 6.52% and for the complete ASDiv dataset it is 5.84%. We get a score of 5.92% for the entire HAWP dataset.

5.2.2. Corpus Lexicon Diversity (CLD)

We also found the corpus lexicon usage diversity metric, CLD as proposed by (Miao et al., 2020). For a given MWP P_i in a dataset P the *lexicon usage diversity* (LD) is defined as:

$$LD_i = 1 - \max_{j, j \neq i} \frac{BLEU(P_i, P_j) + BLEU(P_j, P_i)}{2}$$

where $BLUE(P_i, P_j)$ is the BLEU score (Papineni et al., 2002) between P_i and P_j. The BLEU score is measured with n-grams up to $n = 4$. CLD is given by the mean of all LD_i. This metric ranges from 0 to 1 where a higher value indicates the corpus is more diverse. ASDiv states its CLD as 0.49 while we calculated MAWPS's to be 0.42 and HAWP's CLD was calculated as 0.73.

5.2.3. Reduced Lexical Overlap

To shed some more light on the diversity of problems in a dataset, we used a fixed threshold th to filter problems which were similar to each other. The first step comprised of removing all numeric quantities and punctuation to remove any kind of insignificant diversity. Then we calculated the Jaccard Similarity for each pair of MWPs (unigrams). Table 6 shows the results of filtering lexical overlapping problems using different thresholds for some of the recent benchmark datasets.

Similarity Threshold	MAWPS Reduced Size (Total: 2373)	ASDiv Reduced Size (Total: 2305)	HAWP Reduced Size (Total: 2336)
0.9	1450 (61.10%)	2298 (99.69%)	2259 (96.7%)
0.8	1316 (55.45%)	2274 (98.65%)	2112 (90.4%)
0.7	1179 (49.68%)	2227 (96.61%)	1873 (80.17%)
0.6	1035 (43.61%)	2131 (92.42%)	1503 (64.34%)

Table 6: Reduction of Datasizes after Removal of Similar MWPs

These measures clearly show that HAWP has lexically diverse word problems.

6. Experimental Setup

We pose word problem solving as sequence to sequence learning task. We implemented this by using the open source open NMT (Klein et al., 2017) toolkit.

6.1. Preprocessing

Several preprocessing steps were carried out before passing the data into the openNMT toolkit.

6.1.1. Word to Number Conversion

Many a times numbers are written in a form of words in a word problem. In order to form an equation and thereby finding the correct solution, we need to convert these numbers written in their word equivalents to their corresponding numeric value. We developed an in-house convertor tool to perform this task.

parameter	value
Subword Embedding Size	300
Encoder Layers	2
Decoder Layers	2
Input Sequence Length	200
Output Sequence Length	200
Dropout Rate	0.3
Batch Size	32
Optimizer	Adam

Table 7: Configuration of BiLSTM model with Global Attention for Hindi

6.1.2. Unit Conversion

As a preprocessing step, we normalized quantities related to currency, length, volume, weight, time. When a quantity is described with the help of two co-occurring units, a larger and a smaller one, we normalize them into the larger unit as shown in table 10.

6.1.3. Special Number Token Replacement

In order to reduce the diversity of equation templates, it is a common practice to map actual number into special identifiers (Wang et al., 2017). We replace the numbers appearing in the problem text and equation with symbols from the set $\{p, q, r, s, t, u\}$ with uniform probability similar to (Mishra et al., 2018). The common practice of using of sum_i where $i = 1, 2, 3, .., n$ (n depends on the total number of quantities present in the word problem) was avoided as subword embeddings were used for representing the tokens. The subword model splits sum_i into two tokens num and 0 (0 denotes any one digit number, 00 for two digit number, 000 for three digit number and so on), so single character variables were used for representing the numbers. This mapping between the numbers and special symbols are stored. This strategy of number mapping is used for explicit numbers which are present in the problem text. For implicit numbers like dozen (=12), year (=365 days), week (=7 days), that are required for unit conversion, we also use single character symbols apart from $\{p, q, r, s, t, u\}$ for representing them.

Question	**Implicit Quantity**	**Symbol Used**
agar 1 kele ka mooly 10 rupaye hai, to 1 **darjan** kele ka mooly kitana hoga? *Gloss: If 1 banana costs 10 rupees, then how much would 1 **dozen** bananas cost?*	$darjan = 12$	h
ratan agar 1 din mein 100 rupaye kamaata hai, to 1 **hafte** mein vah kitana kama lega? *Gloss: If in 1 day, Ratan earns 100 rupees, then how much would he earn in 1 **week**?*	$hafte = 7$	g

Table 8: Implicit Quantity Examples

6.1.4. Equation Notation Conversion

All the equations are annotated in the infix notation. Many previous works (Patel et al., 2021; Griffith and Kalita, 2019; Griffith and Kalita, 2021) showed that deep neural architectures perform well when predicting equations in prefix notations. So all the infix notations were converted into corresponding prefix equations.

6.1.5. Conversion into Subwords

The final preprocessing step is the conversion of tokens into their subword forms. We used the BPEmb [2] package using pretrained subword embeddings and subword models to perform this task.

Type	Co-occurring Units	Normalized Unit
Currency	10 rupaye 15 paise	10.15 rupaye
Length	50 meetar 50 senteemeetar	50.50 meetar
Weight	do kilo 300 graam	2.300 kilo
Volume	1 leetar 200 milee	1.200 leetat
Time	2 ghante 45 minit	2.75 ghante

Table 10: Unit Conversion Examples

6.2. Setting

We used the publicly available pre-trained subword embedding (Heinzerling and Strube, 2018) for encoding our Hindi input data. This embeddings are learnt by training on Hindi Wikipedia data using byte pair encoding. We used the same subword embeddings to encode both the word problems and the target equations. We used a 2 layer BiLSTM (Graves and Schmidhuber, 2005) encoder decoder network with global attention (Bahdanau et al., 2014) for predicting the prefix equations given a word problem. The hyper parameter details are shown in Table 7.

7. Evaluation

Most of the mathematical word problem solvers are evaluated either on equation accuracy or solution accuracy. The equation accuracy metrics strictly penalizes any unmatched equation. The solvers do not leverage the equation equivalence property of the generated equations. Here, we introduce the concept of equation equivalence with examples given in Table 15 (Appendix). We also observed that equation equivalence improves the performance of the model by 2% on an average.

8. Results and Discussion

We performed 10 fold cross validation on the whole dataset. The results are shown in table 11. Most of the errors is attributed to incorrect operator identification. The solver also struggles to identify the implicit quantities and could not make correct association with the actual quantity. This is due to very low frequency of such numbers in the word problems. We observed that when problems with implicit quantities are removed from the dataset, the accuracies of the solver increases by approximately 5% on an average. The gain is across the dataset improving both the one operator and two operator equation. Only 2% (64 out of 2336) of the word problems contained implicit quantities. This proves our earlier assertion that the low frequent implicit quantities are harder to learn.

[2] https://github.com/bheinzerling/bpemb

Question	Expected	Predicted	Possible Reason
b bageeche mein paudhon kee t panktiyaan aur r kolam hain. kul kitane paudhe hain? *Gloss: b garden has t rows and r columns of plants. How many plants are there in total?*	t * r	t + r	Requires world knowledge to count the total number for rows and columns within an area
b tikat kee keemat $p hai. c tikaton kee keemat $u hai. d tikaton kee keemat $q hai. yadi har tikat kee laagat samaan hai, to e tikaton kee laagat kitanee hogee? *Gloss: b tickets cost $p. c tickets cost $u. d tickets cost $q. If all tickets cost the same, then how much would e tickets cost?*	p + q	q / p	This is a downside of replacing actual numbers with special tokens.
p tikonon mein kul milaakar kitane kone ho jaenge? *Gloss: How many corners would p triangles have?*	p * d	p + d	Requires world knowledge each triangle('tikona') has 3 (which is the value of d) corners('kone').

Table 9: Examples of Erroneous Cases

	Accuracy
Full Set	34.82
Full Set with No Implicit	39.92
One-Op	40.04
Two-Op	17.81
One-Op with No Implicit	44.43
Two-Op with No Implicit	19.03

Table 11: Average Accuracy after 10-fold Cross Validation

If we analyse the second example in Table 9:

Number replaced by special tokens:	b tikat kee keemat $p hai. c tikaton kee keemat $u hai. d tikaton kee keemat $q hai. yadi har tikat kee laagat samaan hai, to e tikaton kee laagat kitanee hogee?
Original:	1 tikat kee keemat $0.34 hai. 2 tikaton kee keemat $0.68 hai. 3 tikaton kee keemat $1.02 hai. yadi har tikat kee laagat samaan hai, to 4 tikaton kee laagat kitanee hogee?

There are many possible equations for this word problem (not just equivalent equations) namely, $x = p + q$, $x = u * u$, $x = e * p$. However, without the actual numerals it is difficult to predict them. Had there been better representations for numbers for equation generation, predicting the possibility of having multiple equations and those equations might have been easier.

9. Conclusion and Future Work

In this paper, we created a dataset consisting of word problems in Hindi. We also developed baseline neural models for solving word problems. We hope this will enthuse researchers towards this challenging NLP task in low resource languages. The lexical diversity of this dataset is comparable with most of the available benchmark datasets, so it can be used as a benchmark dataset for word problem solving in Indian languages. We also proposed a new evaluation metric leveraging the equivalence property of mathematical equations. As a future work, we will explore different data augmentation techniques to enhance the size of our dataset. For improving the models, we will take up the task of fine-tuning available BERT and other transformer based models.

Acknowledgement

We thank National Council of Educational Research and Training (NCERT), India, Punjab School Education Board, Punjab, India, and Board of Secondary Education, Rajasthan, India for making Hindi word problems publicly available.

10. References

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. (2019). MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bobrow, D. G. (1964). Natural language input for a computer problem solving system. *Ph. D. Thesis, Department of Mathematics, MIT*.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Griffith, K. and Kalita, J. (2019). Solving arithmetic word problems automatically using transformer and unambiguous representations. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 526–532. IEEE.

Griffith, K. and Kalita, J. (2021). Solving arithmetic word problems with transformers and preprocessing of problem text. *arXiv preprint arXiv:2106.00893*.

Heinzerling, B. and Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman, N. (2014). Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, Doha, Qatar, October. Association for Computational Linguistics.

Huang, D., Shi, S., Lin, C.-Y., Yin, J., and Ma, W.-Y. (2016). How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 88, Berlin, Germany, August. Association for Computational Linguistics.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. Association for Computational Linguistics.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. (2016). Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157.

Kushman, N., Artzi, Y., Zettlemoyer, L., and Barzilay, R. (2014). Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. (2017). Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL*.

Miao, S.-y., Liang, C.-C., and Su, K.-Y. (2020). A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online, July. Association for Computational Linguistics.

Mishra, P., Kurisinkel, L. J., Sharma, D. M., and Varma, V. (2018). EquGener: A reasoning network for word problem solving by generating arithmetic equations. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, 1–3 December. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Patel, A., Bhattamishra, S., and Goyal, N. (2021). Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June. Association for Computational Linguistics.

Roy, S. and Roth, D. (2016). Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*.

Roy, S. and Roth, D. (2018). Mapping to declarative knowledge for word problem solving. *Transactions of the Association for Computational Linguistics*, 6:159–172.

Wang, Y., Liu, X., and Shi, S. (2017). Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark, September. Association for Computational Linguistics.

Zhao, W., Shang, M., Liu, Y., Wang, L., and Liu, J. (2020). Ape210k: A large-scale and template-rich dataset of math word problems.

Appendix

Table 12 shows some examples of how guidelines of naturalness, and grammatical correctness were followed while post editing for augmentation:

	English	Direct Translation	After Post editing	Remarks
Naturalness	Molly had 14 candles on her birthday cake. She grew older and got 6 more on her birthday cake. How old is Molly now?	maulee ke barthade kek par 14 momabattiyaan theen. vah badee ho gaee aur usake barthade kek par use 6 aur mileen. seema ab kitanee badee hai?	seema ke janmadin ke kek par 14 momabattiyaan theen. kuchh saalon baad, umar badhane par usane apane janmadin ke kek par 6 aur laga leen. ab seema kee umr kya hai?	More details have been added to the problem and some parts have been changed to make it more natural.
Naturalness	Mrs. Sheridan has 22.0 fish. Her sister gave her 47.0 more fish. How many fish does she have now?	shreematee lata ke paas 22 machhaliyaan hain. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas kitanee machhaliyaan hain?	shreematee lata ke paas 22 machhaliyaan theen. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas ***kul*** kitanee machhaliyaan hain?	"kul" has been added to make the word problem more natural.
Grammatical Correctness	Mrs. Sheridan has 22.0 fish. Her sister gave her 47.0 more fish. How many fish does she have now?	shreematee sheridan ke paas 22 machhaliyaan hain. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas kitanee machhaliyaan hain?	shreematee lata ke paas 22 machhaliyaan theen. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas kul kitanee machhaliyaan hain?	Past tense should be used to describe the state before a change or transaction instead of present tense.
Grammatical Correctness	Isabella's hair is 18.0 inches long. If her hair grows 4.0 more inches, how long will it be?	izaabel ke baal 18 inch lambe the. yadi usake baal 4 inch badhe, ve kitane lambe honge?	gauree ke baal 18 inch lambe the. yadi usake baal 4 inch badhe, ***to*** usake baal kitane lambe honge?	Conditional sentences using 'yadi' (if) require 'to' (then) in Hindi.

Table 12: Examples of Different Guidelines for Augmentation using Translation

Table 13 shows some examples of how guidelines of increasing diversity of MWPs while post editing for augmentation allowed the dataset to include different versions of the same problem to present a test to solvers:

S.No	Problem	Equation	Variation
1.1	raam is maheene 11 kriket ke maich dekhane gaya. vah pichhale maheene 17 maich dekhane gaya tha aur agale maheene vah 16 maich dekhane jaega. vah kul kitane maich dekhega?	X = 11+17+16	Original
1.2	raam is maheene 11 kriket ke maich dekhane gaya. vah pichhale maheene 17 maich dekhane gaya tha aur agale maheene 16 maich dekhane jaane ka soch raha hai. vah kul kitane maich dekh chuka hai?	X = 11+17	Changed Question
1.3	raam is maheene 11 kriket ke maich dekhane gaya. vah pichhale maheene 17 maich dekhane gaya tha aur agale maheene vah 16 maich din mein aur 12 maich raat mein dekhane jaega. vah kul kitane maich dekhane jaega?	X = 16 + 12	Added relevant information and changed question
2.1	raanee mithaee kee dukaan par kaam karatee hai. usane somavaar ko 45 beche. usane mangalavaar ko 16 beche. raanee ne kitane ghevar beche?	X = 45+16	Original
2.2	raanee mithaee kee dukaan par kaam karatee hai. usane somavaar ko 45 beche. usane mangalavaar ko 16 kam beche. raanee ne kitane ghevar beche?	X = 45+(45-16)	Added quantifier

Table 13: Variants MWPs to Increase Diversity of Problems

Table 14 shows examples of errors encountered while augmenting Hindi word problems using Machine Translation.

Type of Error	Example		
	English	Hindi	Explanation
PP-Attachment	Sara picked 45 pears and Sally picked 11 pears from the pear tree. How many pears were picked in total?	saara ne 45 naashapaatee aur sailee ne naashapaatee ke ped se 11 naashapaatee lie. kul kitane naashapaatee chune gae?	'from the pear tree' attaches to both sentences connected by and but in Hindi it goes to only one.
Tense, Aspect, Mood (TAM)	A ship full of grain crashes into a coral reef. By the time the ship is fixed, 49952.0 tons of grain have spilled into the water . Only 918.0 tons of grain remain onboard. How many tons of grain did the ship originally contain?	anaaj se bhara jahaaj moonga chattaan mein durghatanaagrast ho gaya. jab tak jahaaj ko theek kiya jaata hai, tab tak 49952.0 tan anaaj paanee mein gir chuka hota hai. jahaaj par keval 918.0 tan anaaj bacha hai. jahaaj mein mool roop se kitane tan anaaj tha?	wrong TAM due to translation in narrative style
TAM	Kelly has 121.0 Nintendo games. How many does Kelly need to give away so that Kelly will have 22.0 games left?	kelee ke paas 121.0 nintendo gems hain. kelee ko kitane dene honge taaki kelee ke paas 22.0 gem bache hon?	wrong TAM
Fractions or Missing Translataion	Your class had a pizza party. 0.375 of a pizza was left over, and 0.5 of another pizza was left over. You put them both into 1.0 box. How much pizza do you have altogether?	aapakee kaksha mein pizza paartee thee. ek pijja ka 0.375 bacha hua tha, aur doosare pijja ka 0.5 bacha hua tha. aap un donon ko 1.0 boks mein daal den. aapake paas kul milaakar kitana pijja hai?	missing word 'hissa' for Hindi sentence to make sense
Phrasal Verbs	The school cafeteria ordered 42.0 red apples and 7.0 green apples for students lunches. But, if only 9.0 students wanted fruit, how many extra did the cafeteria end up with?	skool kaipheteriya ne chhaatron ke lanch ke lie 42.0 laal seb aur 7.0 hare seb ka ordar diya. lekin, agar keval 9.0 chhaatr phal chaahate the, to kaipheteriya kitane atirikt ke saath samaapt hua?	phrasal verb 'end up' is translated as its constituent verb
Wrong Sense	Luke was putting his spare change into piles. He had 5.0 piles of quarters and 5.0 piles of dimes. If each pile had 3.0 coins in it, how many coins did he have total?	lyook apane atirikt parivartan ko bavaaseer mein daal raha tha. usake paas kvortar ke 5.0 dher aur daims ke 5.0 dher the. yadi pratyek dher mein 3.0 sikke hon, to usake paas kul kitane sikke the?	words with multiple senses are translated with the wrong sense for the given context
Wrong Sense	There are 14.0 rulers and 34.0 crayons in a drawer. Tim takes out 11.0 rulers from the drawer. How many rulers are now in the drawer?	ek daraaj mein 14.0 shaasak aur 34.0 kreyon hote hain. tim 11.0 shaasakon ko daraaj se nikaalata hai. daraaj mein ab kitane shaasak hain?	words with multiple senses are translated with the wrong sense for the given context
Inconsistent Honorifics	Dan found 56.0 seashells on the beach, he gave Jessica some of his seashells. He has 22.0 seashells. How many seashells did he give to Jessica?	dain ko samudr tat par 56.0 seep mile, unhonne jesika ko apane kuchh seeshels die. usake paas 22.0 seeshels hain. usane jesika ko kitane seepiyaan deen?	Inconsistent use of (honorific) third person pronoun

Inconsistent translation of the same word	Olivia gave her cat two cheese cubes. Now Olivia has ninety-eight cheese cubes left. How many cheese cubes did Olivia have originally?	oliviya ne apanee billee ko paneer ke do tukade die. ab oliviya ke paas ninyaanabe paneer kyoobs bache hain. oliviya ke paas mool roop se kitane paneer kyoobs the?	same word 'cube' is translated differently in different statements of the same MWP.
Literal Translation of borrowed words	Fred had 26 chicken wings and gave 18 to Mary. He then finds an unopened box of 40. How many chicken wings does he have in all?	phred ke paas 26 murge ke pankh the aur unhonne mairee ko 18 pankh die. phir use 40 ka ek khula hua dibba milata hai. usake paas kul kitane chikan pankh hain?	Borrowed compound is translated as its constituents
Other	Rachel bought 2.0 coloring books. 1.0 had 23.0 pictures, and the other had 32.0. After 1.0 week, she had already colored 44.0 of the pictures. How many pictures does she still have to color?	raahel ne 2.0 rang bharane vaalee kitaaben khareedeen. 1.0 mein 23.0 chitr the, aur doosare mein 32.0 the. 1.0 saptaah ke baad, usane pahale hee 44.0 chitron ko rang diya tha. use abhee bhee kitanee tasveeren ranganee hain?	Contradictory use of 'baad' and 'pahale hee' making the sentence senseless.

Table 14: Examples of Translation Errors

Table 15 shows some examples of equation equivalence:

Annotated Equation	Equivalents
$X = (a + b) + c$	$X = a + (b + c)$, $X = a + (c + b)$
$X = a + (b - c)$	$X = (a + b) - c$, $X = (a - c) + b$
$X = a - (b + c)$	$X = (a - b) - c$, $X = (a - c) - b$

Table 15: Equation Equivalence Examples