Hindi Word Problem Solving

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

Harshita Sharma
20171099
harshita.sharma@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
October 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Hindi Word Problem Solving” by Harshita Sharma, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Dipti Misra Sharma

To Dada Maa

Acknowledgments

I am grateful to have received support and guidance from various individuals throughout my journey in completing this thesis. I want to express my heartfelt appreciation to these supportive and intelligent beings, without whom this would not have been possible.

First and foremost, to Prof. Dipti M. Sharma, a big THANK YOU for your unwavering support, kindness and encouragement. Your valuable insights and constructive feedback have helped me navigate through the complexities of my research. From the very first class, I was captivated by your genuine enthusiasm for language. You've helped me look at language and learning the way I never thought I could, and this has been one of my biggest learnings here at IIIT, and I will be eternally grateful for it. Thank you for pushing me and not giving up on me and on this research, even when I momentarily tried to escape it. Your door was always open, and your approachability made me feel comfortable seeking guidance whenever needed.

To all my linguistics and NLP professors, your expertise and passion for linguistics and problem-solving have shaped my interest and love for languages. From the bottom of my heart, I thank you all for being excellent educators. I fondly recall our in-class exercises, be it watching videos in class, those fun linguistic games, or those lively discussions on socio-linguistics.

I extend my sincere gratitude to Pruthwik Sir for being an exceptional mentor. Your mentorship and the calm confidence and patience it accompanies have been invaluable. I am thankful for all the advice and wisdom that you've shared with me every step of the way. You not only guided me to the right path but, first, also encouraged me not to be afraid of making a mess. I am deeply grateful for your countless hours mentoring and supporting me throughout this journey.

I am forever indebted to my family for their unconditional love and support and well, their constant reminders and questions regarding the completion of this thesis. Your sacrifices and understanding have allowed me to pursue my dreams. Thank you for trusting me and always giving me the healthy space to work on myself by myself - falling and getting up repeatedly while also being there whenever I reach out for help and support.

I am thankful to all my seniors and friends, especially Sachin sir, Devansh sir, Mayank Sir, Arjun Sir, Ajju, Sagrika, Atirek and most importantly, the entire CLD gang, for being my constant motivation and joy. Your companionship has made this academic journey not just bearable but truly enjoyable. Your kindness, intelligence and friendship have taught me a lot and made me a better person, and I cherish the memories we have created together more than I can ever express.

I owe all of you a debt of gratitude that words cannot adequately express. Each of you has played a significant and unique role in my personal and academic growth, and I am blessed to have you in my life. Your presence and support have made a considerable difference, and your kindness and generosity truly humble me. Thank you for being a part of my life's journey and making it richer and more meaningful. I look forward to cherishing these connections and continuing to learn and grow with your guidance and support.

Abstract

Word problem Solving is a popular NLP task that deals with solving mathematical problems described in natural language. Mathematical Word Problems cover problems over a large mathematical domain with various complexities ranging from Arithmetic and Algebraic to Geometry and Calculus. While most word problems are entirely textual, some word problems, like geometric word problems, may also have a visual component.

Much research has been carried out to solve different genres of word problems with various complexity levels in recent years. However, most publicly available datasets and work are in English. Recently there has been a surge in word problem-solving in Chinese with the creation of large benchmark datasets. Labelled benchmark datasets for low-resource languages are very scarce.

The first requirement for solving word problems, like any other problem, is data. To the best of our knowledge, no datasets are available for any Indian Languages for Word Problem Solving. Such limitations on data availability not only encouraged us to create a new dataset for an Indian Language but also made us explore techniques by which data for other Indian Languages can be created with ease.

In this work, we present a diverse dataset containing 2336 Arithmetic word problems in Hindi built by manually crafting word problems and using word problems augmented from benchmark datasets of other languages. For augmentation, we used translation of word problems (of other languages in which Word Problem Solving data was developed) as a tool to generate diverse word problems. In this process, we studied the translated word problems, gathered the patterns of issues seen in these translations and defined the steps to eliminate the errors and improve the quality of the translated output for it to be suited to be a set of Hindi word problems that look as natural as word problems that are studied across India in Hindi medium schools.

We also developed baseline systems for solving these word problems - a rule-based solver that uses verbs to identify operations for generating the answers to word problems and an end-to-end deep learning-based solver that generates equations for word problems. We also propose a new evaluation technique for word problem solvers taking equation equivalence into account. This will form the basis for future work for Word Problem Solving in Indian Languages, especially in Hindi.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) has a lot of interesting tasks and problems. Word Problem Solving (WPS) being one of them, mainly focuses on natural language understanding and processing. The task aims to give systems the ability to understand and solve mathematical word problems in much the same way as we humans do.

Math word problems (MWPs) can be defined as descriptions of one or more scenarios describing a mathematical problem, solving which requires the application of mathematical concepts on the numeral values (may also include text) presented in the problem. These mathematical concepts can range from simple addition to complex algebra or calculus. Some examples of word problems are given below:

- Lillian was playing basketball. 2 of her shots went in the hoop. 3 of her shots did not go in the hoop. How many shots did she take in total?
- Jay's father is twice as old as Jay. In 20 years Jay will be two-thirds as old as his father. How old is Jay now?
- A rock is dropped into the center of a circular pond. The ripple moved outward at 4 m/s. How fast does the area change, with respect to time, when the ripple is 3m from the center?

Math Word Problems, mainly taught in primary and high schools, are considered an integral part of the 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. Most often MWPs are designed as real-world narratives. For a machine to correctly model this type of information, form relationships among the quantities and generate the answer is complex.

1.1 Current State of Word Problem Solving

Word problem solving continues to be a challenging task in NLP. Recent works have shown that solving elementary-level word problems itself still poses a significant challenge for the NLP community.

Although difficult, a lot of work has been carried out to create systems that can solve word problems of varying difficulty with a number of innovative and intelligent perspectives: semantic parsing, template matching, explainable solvers etc. One of the primary requirements for these solvers is good-quality data. As a result, a variety of datasets have also been created along with different types of solution approaches.

1.1.1 Datasets

Several large scale datasets have been released over the years for Mathematical word problem solving like AQuA [17] containing 100K complex MWPs and MathQA [1] containing 37K word problems in English and Chinese dataset Ape210K [33] containing 210K problems and 56K templates. Recently, the focus has shifted from large sized datasets to more diverse datasets. [20] 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. [23] introduced a challenge dataset SVAMP for which the best accuracy of state-of-the-art solvers is much lower. [27] also showed that the benchmark datasets are biased and include word problem with high lexical overlapping. Chapter 2 talks more on WPS datasets, their structure and their development in various languages.

1.1.2 Solvers

Solving arithmetic word problems has been fascinating the world of NLP since the 1960s [3] as it involves an interesting yet complicated blend of natural language understanding, identification of relevant and irrelevant quantities, operations as well as semantic reasoning across sentences. Initial solvers were rule based or 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 templates. KAZB [16] was the first attempt to learn these alignments for a set of linear equations. Other statistical approaches [10] used verb categorization for solving word problems where verbs triggered the flow of quantities between entity driven containers. [25] 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 [11] outperformed its sophisticated statistical counterparts on large datasets. [31] 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, [23] shows 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.

1.2 Word Problem Solving in Hindi

Similar to English, Hindi word problems are a part of the school curriculum in many states of India. There are more than 20 education boards in India, out of which many follow a Hindi-medium curriculum; hundred-thousands of Hindi-medium schools in India, and therefore, even more, Hindi studying and teaching members who learn and teach how to solve mathematical word problems almost every day. Thus, the need for word problem solving also exists in Hindi and other Indian Languages (ILs).

As mentioned earlier, 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. We have created a dataset for Hindi Word Problem Solving to address this. To our knowledge, this is the first dataset for arithmetic word problems in any Indian language. We adopted a two-pronged strategy to construct a diverse and challenging dataset of 2336 MWPs. Some of these MWPs were collected from Hindi textbooks, worksheets, and Hindi-medium educators, while most were augmented using translation. Through this thesis, we make the following contributions:

- We tackled the problem of a lack of Word problem solving datasets and solvers in the Hindi language.
- We designed a good quality, diverse and challenging, publicly available[1] dataset of 2336 Arithmetic MWPs annotated with equations, number of operations and indices of relevant quantities in the word problems.
- We propose baseline systems and equation equivalence techniques to handle multiple possibilities for equations.

1.3 Organisation of the Thesis

In this section, we define the structure of this thesis and present a summary for each of the chapters:

- **Chapter 1: Introduction** describes the problem of word problem solving and presents the overview of the current state of research surrounding Word Problem Solving in Hindi and other languages.
- **Chapter 2: Dataset for Hindi Word Problem Solving** presents the process of creation of a Hindi dataset for word problem solving. This chapter also describes the structure of WPS datasets and presents an overview on different WPS datasets available.

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

- **Chapter 3: Augmentation of Hindi Word Problems** talks about how data augmentation was used to create more Hindi word problems, the challenges faced, and how we resolved them.
- **Chapter 4: Nature of Dataset** states the results of the evaluation of the dataset on two properties: naturalness and diversity.
- **Chapter 5: Hindi Word Problem Solver: Verb Categorisation for Word Problem Solving** describes the Verb Categorisation solver. For the task of verb categorisation, we explore 2-3 different approaches and finally use verb categories to solve word problems using a rule-based system.
- **Chapter 6: Hindi Word Problem Solver: Deep learning Approaches for Word Problem Solving** describes an end-to-end deep learning-based solver. This chapter also introduces and explains equation equivalence techniques that can be used while evaluating WPS models.
- **Chapter 7: Conclusion and Future Work** concludes this thesis. We present a consolidated view of the results achieved through the work done in this thesis and also lay down possible future avenues of research for using our dataset to develop word problem solving systems or to create more datasets for low-resource languages.

Chapter 2

Dataset for Hindi Word Problem Solving

In this chapter, we look at the structure and composition of Math Word Problem (MWP) datasets with a few examples from English and Chinese datasets. We also highlight the issues in current MWP datasets. While considering these issues, we describe the process used to create a new diverse dataset for Hindi Word Problem Solving.

2.1 Math Word Problem(MWP) Datasets

2.1.1 Properties and structure of an MWP dataset

To understand the structure of an MWP dataset and the importance of its constituents, let's have a look at one of the WPS datasets first. Figure 2.1 provides a peek into how ASDiv [20] dataset is structured.

Now, keeping the above snapshot as a reference, let us look at the following properties and constituents of a MWP dataset:

- **Word Problems or MWPs:** Word problems as defined in Chapter 1. Some datasets list the entire word problem (as shown in Fig. 2.2) while others separate the statements or the body of the word problem and the question (as shown in Fig. 2.1).
- **Annotation:** Extra details or information about the word problems. MWP datasets commonly have word problems annotated with Equations (sometimes called 'Formula') and Answers. Sometimes, more insight is provided through annotations like Operation Type, Problem Type, Quantities, and Grade level.
  - **Equation:** refers to the mathematical equation generated from a word problem which can be solved to get the answer to the word problem. This is the most crucial annotation because it acts like the target variable whose value is to be predicted and is used to test the accuracy of word problem solvers. It must be noted that 'Answer' can also be treated as a target variable. However, checking whether a system has clearly understood the problem and can

```
<?xml version="1.0" encoding="UTF-8" ?>
<Machine-Reading-Corpus-File>

<ProblemSet>
        <Problem ID="nluds-0001" Grade="1" Source="http://www.k5learning.com">
                <Body>Seven red apples and two green apples are in the basket.</Body>
                <Question>How many apples are in the basket?</Question>
                <Solution-Type>Addition</Solution-Type>
                <Answer>9 (apples)</Answer>
                <Formula>7+2=9</Formula>
        </Problem>
        <Problem ID="nluds-0002" Grade="1" Source="http://www.k5learning.com">
                <Body>Ellen has six more balls than Marin. Marin has nine balls.</Body>
                <Question>How many balls does Ellen have?</Question>
                <Solution-Type>Addition</Solution-Type>
                <Answer>15 (balls)</Answer>
                <Formula>6+9=15</Formula>
        </Problem>
        <Problem ID="nluds-0003" Grade="1" Source="http://www.k5learning.com">
```

Figure 2.1: Snapshot of ASDiv [20] dataset

solve by following the expected steps can be done only by looking at the 'Equation'. The 'Answer' alone does not give us any information about the calculations and steps involved while solving a word problem.

– **Answer:** refers to the solution to the question asked in the word problem, attained after extracting and solving its equation. Some models/solvers use the answer as a target variable to check their accuracy.

– **Quantities:** refers to all the numerals or number words (numbers in word form, e.g. twenty, fifty-nine, ten etc.) that are mentioned in the word problem. Datasets which do not use number words, do not consider 'a' which also denotes the number 'one' as a quantity. For example, we can see in the following snapshot of a word problem from Unbiased dataset [27] that it does not include '1' as a quantity denoting the number of cakes being baked by Mary:

```
{
  "iIndex": 3,
  "sQuestion": "Mary is baking a cake . The recipe wants 8.0 cups of
  flour . She already put in 2.0 cups . How many cups does she need to
  add ?",
  "quants": [
    8.0,
    2.0
  ],
```


Figure 2.2: Example of Annotation type: Quantity

- **Alignment:** specifies the indices of quantities in the word problem in the order that they occur in the equation. It is clear that alignment only refers to quantities required to solve the problem. This information helps the solvers differentiate between relevant and irrelevant quantities in word problems.
- **Operation Type:** specifies the name of the arithmetic operation required to solve the word problem. This helps solvers understand and build a relation between operation type and word problem description, which can help predict the operation in the equation.
- **Problem Type:** specifies the name of the branch of mathematics to which the problem in the dataset belongs, like Number-word problems, Arithmetic, Algebra, Geometry, Calculus etc.
- **Grade Level:** used to mark the level of difficulty of a problem.

Annotations like Grade Level and Problem Type can be used while evaluating the quality and diversity of the dataset. They also shed more insight while making inferences from solver results and other dataset experiments.

Examples of word problems and their annotation covered by such datasets can be found in Table 2.1.

The properties of an MWP dataset are its size and language:

- **Dataset size:** refers to the number of unique word problems in the dataset. Some word problems may be similar or even the same in meaning but slightly differ in structure, while some problems may have the same statements, and the question following the statements may vary. More on why such word problems are considered unique and why they are important can be found in Chapter 3. Like any other domain, in WPS too, the bigger the dataset, the more useful it is. However, the quality of a dataset also holds equal importance; thus, the dataset quality must not be traded off for size.

Problem	Equation	Answer	Operation	Problem Type	Grade Level
Mr. Fortree, a businessman also gave seedlings for the tree planting activity. If he gave 14 seedlings of cedar and 38 seedlings of pine, how many seedlings did he give in total?	14+38=52	52	Addition	Arithmetic	3
There are 43 students and 1720 apples. Each student has 9 Skittles. If the apples are divided equally among the students, how many does each student get?	1720/43=40	40	Common-Division	Arithmetic	4
Bryan took a look at his books as well. If he has 56 books in each of his 9 bookshelves, how many books does he have in total?	56*9=504	504	Multiplication	Arithmetic	3

Table 2.1: Examples of Word Problem Annotation (Dataset: ASDiv)

- **Language:** This refers to the language in which the word problems are defined. The word problems in a dataset may or may not be written in the script of the language it is defined in. Most datasets include numerals in English, but some may consist of numerals of their respective scripts.

Now that we have a good understanding of the composition of the WPS datasets along with the importance of each constituent, let us look at real-world MWP datasets that are being used to create solvers or to create new improved datasets.

2.1.2 MWP Datasets in English

Table 2.2 presents some of the well-known MWP datasets available in English. These datasets cover different MWP categories ranging from Arithmetic to Algebraic to MCQ/Fill in the blanks type of questions. These datasets also have word problems of varying linguistic and mathematical complexity.

Dataset	MWP Category	Annotation	Size	Year
AI2 [10]	Arithmetic word problems	Equation, Answer	395	2014
KAZB [16]	Algebraic word problems	Equation, Answer	514	2014
DRAW [29]	Algebraic word problems	Equation, Answer, Template	1000	2015
IL [24]	Arithmetic word problems	Equation, Answer	562	2015
ALGES [14]	Algebraic word problems	Equation, Answer	508	2015
Dolphin1878 [28]	Number-word problems	Equation, Answer	1878	2015
Dolphin18k [11]	Arithmetic, Algebraic, Domain knowledge problems	Equation, Answer	18000	2016
AllArith [26]	Arithmetic word problems	Equation, Answer	831	2017
AQuA [18]	Arithmetic, Algebraic, Domain knowledge problems	Rationale, Answer	100000	2017
Unbiased [27]	Arithmetic word problems, Domain Knowledge problems	Equation, Answer, Quantities, Alignment	1492	2018
MathQA [1]	Arithmetic, Algebraic, Domain knowledge problems	Decomposed linear formula, Answer	37000	2019
ASDiv [20]	Arithmetic, Algebraic, Domain knowledge problems	Equation, Answer, Problem Type, Grade level	2305	2020
SVAMP [23]	Arithmetic word problems	Equation, Answer	1000	2021

Table 2.2: English MWP datasets

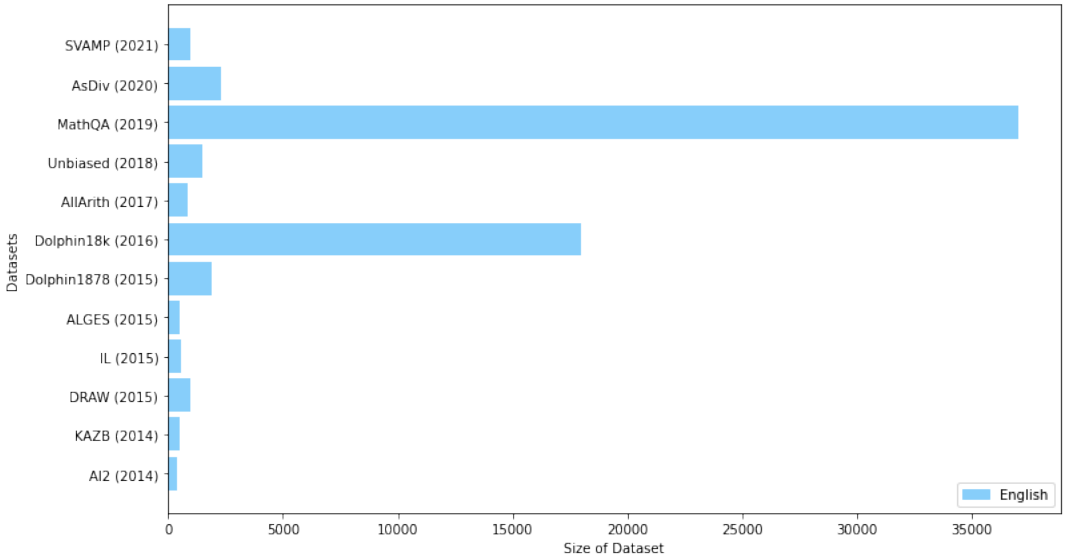


Figure 2.3: Sizes of various English datasets

Figure 2.3 provides a visual representation to understand the differences in the size of these English datasets. The x-axis denotes the size of the dataset, while dataset names are listed on the y-axis. Therefore, the length of horizontal bars depicts the dataset size for the dataset name written against it on the y-axis.

The two extremes on the graph: MathQA has 37000 problems, and AI2 has 395 problems. While some datasets like MathQA and Dolphin18k have exceptionally high numbers of problems, most English datasets have around 1000-3000 word problems.

2.1.3 MWP Datasets in Other Languages

Apart from English, some research has also been carried out on Chinese word problem solving. The Chinese datasets are enormous. The most well-known datasets are presented along with their sizes in a graph in Fig. 2.4. The length of horizontal bars depicts the dataset size for the dataset name written against it on the y-axis.

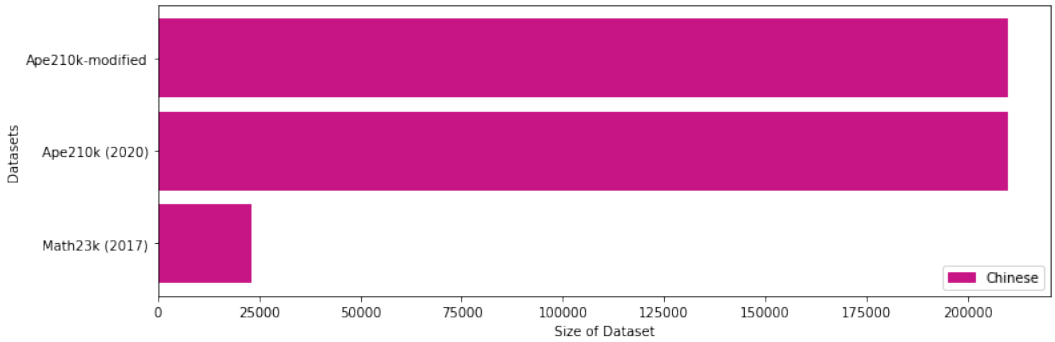


Figure 2.4: Sizes of available Chinese datasets

If we put the two graphs of English and Chinese dataset sizes together in Fig. 2.5, it becomes clear that Chinese has some of the most extensive MWP datasets in front of which even one of the largest English datasets, AQuA containing 100,000 word problems looks small.

As seen above, a large amount of data for WPS exists in English and Chinese, covering different problem categories. While abundant data is available through these datasets, current datasets have some issues that directly or indirectly affect the dataset quality and hence the solvers. These issues are discussed next.

2.1.4 Issues in Current MWP Datasets

While discussing the properties of an MWP dataset, it was pointed out that trade-offs between quantity and quality should be made very carefully. Some datasets have issues that damage their quality, making them less suitable for WPS, despite their large size. Some of the problems which were observed in current MWP datasets are discussed below:

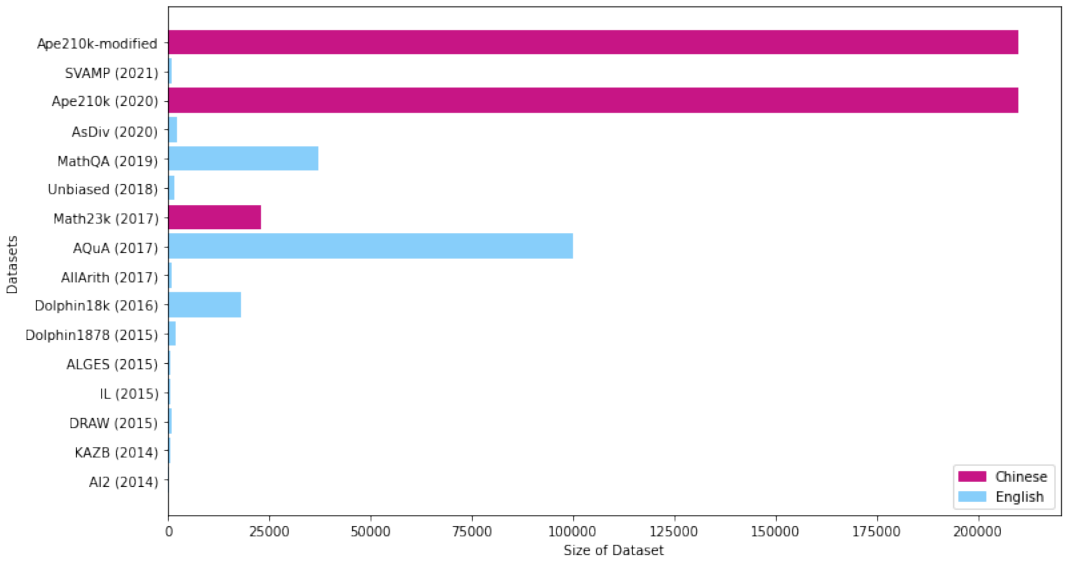


Figure 2.5: Comparing English and Chinese dataset sizes

2.1.4.1 Near-Duplicate Problems

As [1] points out, AQuA [18], one of the largest English MWP datasets, contains a lot of redundant problems with slight changes in proper nouns and numerals. This issue is not specific to AQuA and can also be found in many other datasets. Table 2.3 shows some of the problems which prove this argument.

2.1.4.2 Errors in Annotation

Many large datasets have been developed through crowd-sourcing and contain errors. Most of these datasets are expanded from a small set of seed problems by altering a few keywords and numerals. While the word problems are altered, these changes are not made in the equations and answers. This leads to cases when the equation for an MWP is incorrect. Some examples of errors in annotation are given in Table 2.4.

2.1.4.3 Less Diverse Datasets

Due to the occurrence of the same sentence pattern in MWPs and because of near duplicate problems, the lexical diversity of datasets is reported to be very low. [20] reports the Lexicon Usage Diversity for the following datasets:

Dataset	Dataset Size	Lexical Diversity
AI2	395	0.30
IL	562	0.27
KAZB	514	0.39
ALGES	508	0.35
DRAW	1000	0.35
AllArith	831	0.40
MathQA	37K	0.05
ASDiv	2305	0.49

2.2 Hindi Dataset Construction

Building a rich dataset for a low-resource language like Hindi is complex, especially since no MWP repository is available. Though MWPs are a crucial part of the Math curriculum in Hindi-medium schools and examinations, finding diverse data in reasonable quantities is difficult. That being the case, we constructed a dataset, HAWP (Hindi Arithmetic Word Problems), of 2336 word problems using these two methods:

- **Manually Crafted MWPs:** A total of 736 word problems were crafted manually. The process of how this was carried out will be discussed further in this section in more detail.
- **Data Augmentation:** A total of 1600 word problems were generated using data augmentation technique. Chapter 3 provides a comprehensive view on the same.

Fig. 2.6 shows the distribution of word problems created using these two methods.

2.2.1 Manually Crafted MWPs

Manually crafted MWPs, as the name suggests, are math word problems that were manually formulated. As figure 2.6 shows, 32% of HAWP was created using this method. We needed some benchmark data that students and teachers are using in the real world. Therefore, we created at least some portion of the dataset manually that would also serve as a full-proof set of word problems. This was done with the help of Hindi-medium Math teachers and Hindi-medium Math textbooks that are currently being used across India by various education boards.

Figure 2.7 shows the distribution of MWPs that were created manually. 25% of the word problems in the HAWP dataset were crafted using Hindi-medium Math textbooks (i.e. around 80% of all manually crafted problems in HAWP) while 6% of HAWP was created by Hindi-medium Math teachers.

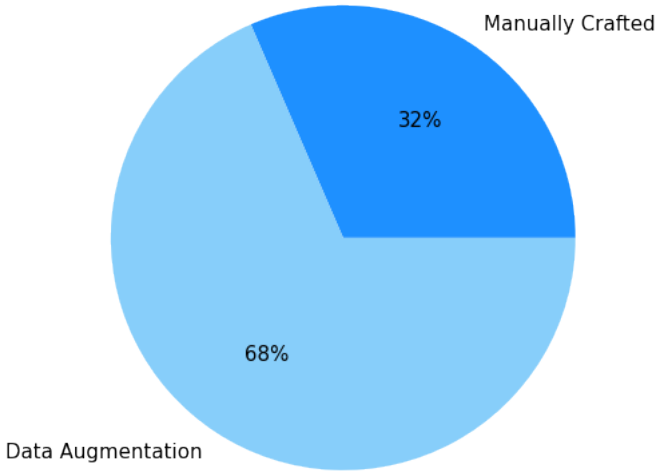


Figure 2.6: Distribution of MWPs generated using two methods

2.2.1.1 Hindi-medium Math Teachers

To create Hindi word problems, we contacted two Math teachers from Hindi medium schools responsible for formulating examination papers. These Math teachers from Hindi-medium schools were asked to develop word problems of 1 to 6 grade level as a worksheet or exam they would make for their students to collect MWPs that are relevant in the real world and used by students and teachers.

To match the MWP type of the dataset being created and to get consistency and diversity in the kind of problems, the teachers were given instructions indicating that the crafted word problems:

- must be of arithmetic level comprising of calculations using any of the four operations: addition, subtraction, multiplication and division.
- must be solvable using linear equations only.
- must have only one unknown.
- must have either one or two operation calculations only.
- may or may not have irrelevant information to confuse the solvers.

These points were clarified by giving a few examples of the kind of MWPs we were looking for. It was also notified to the teachers that they should **not** stick to the pattern of examples given to them and meet the criteria mentioned above.

Around 150 word problems were collected from teachers. Two NLP Masters students verified that these problems were logically coherent and followed all the constraints mentioned above.

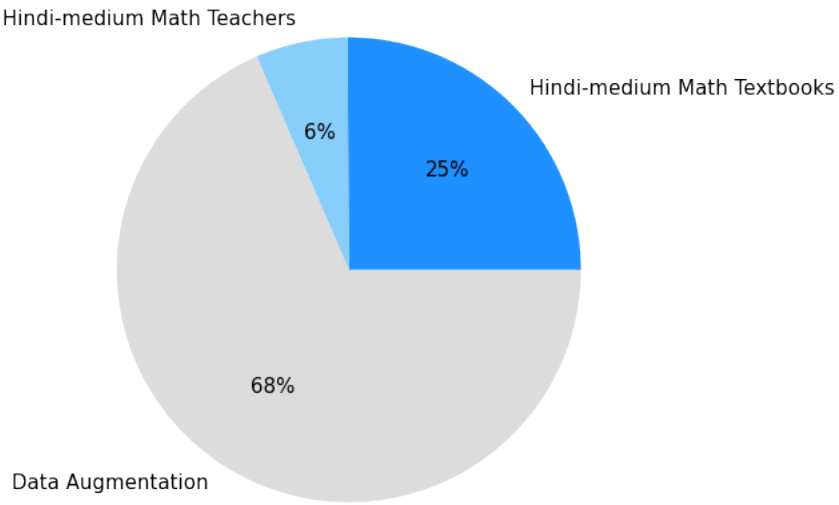


Figure 2.7: Distribution of MWPs that were generated manually

2.2.1.2 Hindi-medium Math Textbooks

We went through many publicly available Hindi-medium Math textbooks and workbooks for grades 1 to 6. Hindi word problems were constructed similar to the textbooks so that the naturalness of the word problems could be maintained.

Looking at these textbooks and workbooks not only familiarised us with standard Hindi Math word problems but also helped us look at a wide variety of problems and break free from the limited view of word problem styles that we usually see in the already available datasets in English. Moreover, it also allowed us to understand the basic style of MWPs that is followed in Hindi, i.e. how specific keywords are frequently used to make the word problem sound more natural, whether the MWPs define the problem very explicitly or leave it a little vague. All these pointers helped us keep problems in the entire dataset sound naturally "word-problem-y" in Hindi.

The data created using these methods was reliable but not enough to be used to train modern systems. We needed more good quality data to be used efficiently by researchers and models, similar to the previously mentioned English and Chinese datasets. Since benchmark data was available in other languages, we decided to make use of that publicly available data by translating word problems from these datasets.

2.2.2 Data Augmentation using Translation

Around 1600 problems from different benchmark datasets in English, namely AI2 [10], Unbiased [27], ASDiv [20] were translated for augmenting MWPs in Hindi. Chapter 3 provides an in-depth description of how problems from these English datasets were translated as a part of data augmentation.

Table 2.5 summarises the number of problems generated using all the methods mentioned above.

Method	#MWPs
Manually Crafted: Hindi-medium Math teachers	150
Manually Crafted: Hindi-medium Math textbooks	586
Augmentation	1600
Total	2336

Table 2.5: Various methods of dataset construction

Once we had a good number of problems in our dataset, we needed the data to be correctly annotated so that it could be used by anyone and everyone to train and test models. Next, we look at the different types of annotations added to the word problems and how the process of doing so was simplified.

2.2.3 Annotation

As discussed in the first section of this chapter, annotation plays a very important role in any WPS dataset. Many different types of annotations are used for MWPs, the most important and mandatory one being the equation/formula corresponding to each word problem.

2.2.3.1 MWP Annotation Types

The crafted and the augmented data was annotated with equations, number of operations and indices of relevant quantities in the word problems (also known as alignment). The meanings of these annotations are recapitulated along with some examples:

- Equation: A mathematical formula containing numerals from the word problem that generates the answer to the word problem.

- Number of Operations: As the name suggests, states the number of mathematical operations to be performed to solve the word problem or the equation. Note: This does not mean unique operations used to solve the problem but total number operations. So, if a problem requires two addition operations, the number of operations for that problem would still be 2.

- Alignment: Indices of relevant quantities in the word problems that are used in the equation.

No.	Problem	Equation	Alignment	#Operations
1	ojasvee ke khaate mein ₹12.08 the aur usane ₹9.85 kharch kie. usake paas kitane paise bache? *Ojaswi had ₹12.08 in her account and she spent ₹9.85. How much money does she have left?*	X = 12.08 - 9.85	[0, 1]	1
2	daraaj mein 33 pensilen aur 44 kreyon the. jaagrti ne 27 pensilon ko daraaj mein rakha. ab kul kitanee pensilen hain? *A drawer has 33 pencils and 44 crayons. Jagruti kept 27 pencils in the drawer. How many pencils are there now?*	X = 33 + 27	[0, 2]	1
3	aneeta ko mithaee banaane ke lie phalon kee 26 petiyon kee zaroorat hai. usake paas pahale se hee 10 petee aam aur 9 petee kele hain. aneeta ko aur kitanee petiyaan chaahie? *Anita needs 26 boxes of fruits to make sweets. She already has 10 boxes of mangoes and 9 boxes of bananas. How many more boxes does Anita need?*	X = 26 - (10 + 9)	[0, 1, 2]	2

Table 2.6: Annotation examples from HAWP dataset

Each example given in Table 2.6 is explained below:

- Example 1: Ojaswi had ₹12.08 in her account and she spent ₹9.85. How much money does she have left?
  - Equation: Amount left = Amount in bank originally - Amount spent
    ⇒Money left = 12.08 - 9.85

- Alignment: The quantities appear in the problem in the order: 12.08 (Index 0), 9.85 (Index 1). Both quantities are required to solve the problem. Hence the alignment is the indices of these two quantities: 0, 1.
- Number of Operations: Clearly from the equation, we can see that one mathematical operation is required to solve the problem.

- Example 2: A drawer has 33 pencils and 44 crayons. Jagruti kept 27 pencils in the drawer. How many pencils are there now?
  - Equation: No. of pencils in drawer now = No. of pencils in drawer originally + No. of pencils kept in drawer by Jagruti
    $\Rightarrow$No. of pencils in drawer now = 33 + 27
  - Alignment: The quantities appear in the problem in the order: 33 (Index 0), 44 (Index 1), 27 (Index 2). Out of these 3 quantities, only two i.e. 33 and 27 are required to solve the problem as they point to the number of pencils and therefore, appear in the equation. Hence the alignment is the indices of these two quantities: 0, 2.
  - Number of Operations: One mathematical operation is required to solve the problem.

- Example 3: Anita needs 26 boxes of fruits to make sweets. She already has 10 boxes of mangoes and 9 boxes of bananas. How many more boxes does Anita need?
  - Equation: No. of more fruit boxes required = No. of fruit boxes required - No. of fruit boxes present
    $\Rightarrow$No. of more fruit boxes required = No. of fruit boxes required - (No. of mango boxes present + No. of banana boxes present)
    $\Rightarrow$No. of more fruit boxes required = 26 - (10 + 9)
  - Alignment: The quantities appear in the problem in the order: 26 (Index 0), 10 (Index 1), 9 (Index 2). All 3 quantities are required to solve the problem. Hence the alignment is the indices of these two quantities: 0, 1, 2.
  - Number of Operations: Two mathematical operations are required to solve the problem.

Manually annotating a large number of problems can be tedious and can lead to errors in annotation. To facilitate easy annotation of equations and the relevant indices, we also developed an equation annotation tool.

2.2.3.2 Equation Annotation Tool

The annotation of number of operations and relevant quantities was done in house by the authors. In order to 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.

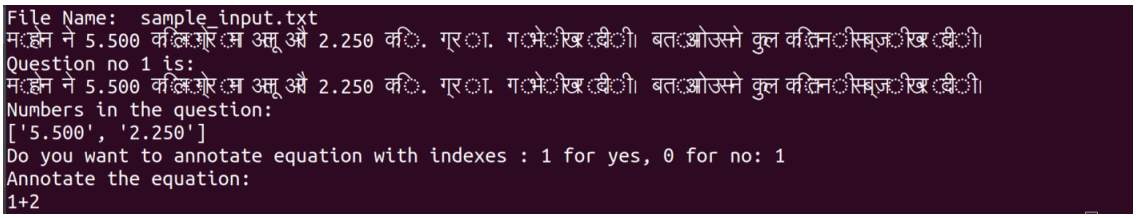


Figure 2.8: Snapshot of the Equation Annotation Tool

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.
Gloss:	Mohan bought 5.500 kilograms of potatoes and 2.250 kg of cauliflower. Find the total weight of the vegetables that he bought.
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

2.2.3.3 Inter-Annotator Agreement

Two annotators with prior experience in automatic word problem-solving were involved in this task. 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 two types. The first kind of disagreement occurred due to the incorrect identification of operands or operations. The other one 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?
Gloss:	Reena buys 1.400 kg of tomatoes and 750 grams of chillies from the market and keeps them in her bag. What is the total weight of her bag?
Equation 1:	$X = (1.400 + (750/1000))$ in kigra or kg
Equation 2:	$X = (1400 + 750)$ in gram

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.

For Type 1 disagreements, we corrected the operations and quantities in the equations whereas for Type 2 disagreements since both are annotations were correct, we kept the annotations as is. However, the Type 2 disagreement bring to light a very important factor of role of unit conversion in answers/equations and this can be incorporated in annotations by including multiple equations/answers in future work. For the scope of this thesis, we have kept only one equation per word problem.

Problem	Equation	Dataset
Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?	x + 1.15x = 43	AQuA
Two friends plan to walk along an 18-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 25% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?	x + 1.25x = 18	AQuA
Two friends plan to walk along a 36-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 25% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?	x + 1.25x = 36	AQuA
Two friends plan to walk along a 33-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 20% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?	x + 1.2x = 33	AQuA
Two friends plan to walk along a 22-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 20% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?	x + 1.2x = 22	AQuA
The school is planning a field trip. There are 45 students and 9 seats on each school bus. How many buses are needed to take the trip?	X = 45 / 9	IL
The school is planning a field trip. There are 14 students and 2 seats on each school bus. How many buses are needed to take the trip?	X = 14 / 2	IL
The school is planning a field trip. There are 9 students and 3 seats on each school bus. How many buses are needed to take the trip?	X = 9 / 3	IL
The school is planning a field trip. There are 28 students and 7 seats on each school bus. How many buses are needed to take the trip?	X = 28 / 7	IL
The school is planning a field trip. There are 180 students and 60 seats on each school bus. How many buses are needed to take the trip?	X = 180 / 60	IL

Table 2.3: Examples of Near-Duplicate Problems

Problem	Annotation	Dataset	Error
A train covers a distance of 10km in 10 min. If it takes 6 sec to pass a telegraph post, then the length of the train is?	**Options:** A)m, B)m, C)m, D)m, E)m **Rationale:** Speed = (10/10 * 60) km/hr = (60 * 5/18) m/sec = 50/3 m/sec. Length of the train = 50/3 * 6 = 100 m. ANSWER:C	AQuA	Incorrect options
A batsman makes a score of 64 runs in the 16th innings and thus increased his average by 3. Find his average after the 16th inning?	**Options:** A)17, B)21, C)22, D)23, E)24 **Rationale:** Let the average after the 16th inning be P. So, the average after the 15th inning will be (P-3). Hence, 15(P-30) + 62 = 16P $\Rightarrow$P = 17. ANSWER:A	AQuA	Incorrect options and Rationale

Table 2.4: Examples of errors in annotation in datasets

Chapter 3

Augmentation of Hindi Word Problems

In this chapter, the entire process of augmentation of Hindi word problems will be explained: from choosing the apt augmentation technique to augmenting data to presenting the challenges faced during augmentation and how these challenges were tackled.

3.1 Data Augmentation Techniques

Data Augmentation is the technique of increasing the amount of data by creating new data using the data available at hand. Multiple data augmentation techniques are being used widely in the field of NLP. They can be applied to different levels in the dataset by adding/removing/changing various elements of the given text, i.e. characters, words, sentences, paragraphs etc. (depending on the type of data). Since WPS data has a group of related sentences as one unit of the dataset, we can use augmentation techniques for character, word, and sentence levels.

Moreover, to augment WPS data, we need the newly generated data to be semantically and syntactically correct and natural while also being structured (the order of statements is important) and solvable word-problem-wise while preserving their equations and other metadata. These constraints are what make data augmentation for MWPs a challenging task.

Techniques like noise injection, deletion, insertion and swapping of words or sentences cannot be used for our problem as they change may make the word problem semantically or syntactically incorrect, alter or distort the equation, or change the structure of the word problem, making it unsolvable. Hence, the various augmentation techniques available in NLP are narrowed down as they may not be helpful for our problem statement.

3.1.1 Data Augmentation Techniques for Math Word Problems

Now that we have a clear problem statement and constraints for data augmentation, going forward our focus will be on augmentation techniques that are useful for WPS data i.e. Math Word Problems.

We also discuss the advantages and disadvantages of each technique that can help find the most suitable technique for the task in this and further sections.

3.1.1.1 Word-level Substitution

The methods used in word level substitution identify and replace some key words of a particular category from the word problem with other keywords belonging to the same category:

- **Person Names' Substitution:** One of the easiest methods in substitution is to replace the Personal Nouns in the word problem. As long as the consistency in the replacement is maintained, there are no other complexities involved in this method. Examples can be found below:

  - **Original:** Alyssa's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with?

    After Substitution: Mandy's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with?

  - **Original:** There are 2 pencils in the drawer. Tim placed 3 pencils in the drawer. How many pencils are now there in total?

    After Substitution: There are 2 pencils in the drawer. Tom placed 3 pencils in the drawer. How many pencils are now there in total?

- **Number Substitution:** The quantities in the word problems are replaced by other numerals. For example:

  - **Original:** Alyssa's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with ?

    After Substitution: Alyssa's dog had puppies. She gave 10 to her friends. She now has 2 puppies. How many puppies did she have to start with ?

  - **Original:** There are 2 pencils in the drawer . Tim placed 3 pencils in the drawer . How many pencils are now there in total ?

    After Substitution: There are 3 pencils in the drawer . Tim placed 7 pencils in the drawer . How many pencils are now there in total ?

  Though this seems like an easy and straightforward approach, this can sometimes lead to incoherence in word problems. For example:

  - **Original:** Alyssa's dog had **15** puppies. She gave some to her friends. She now has 5 puppies. How many puppies did she give to her friends?

    After Substitution: Alyssa's dog had 10 puppies. She gave some to her friends. She now has 25 puppies. How many puppies did she give to her friends?

Problem: As we can see, the number of puppies after giving away some of them is more than the number of puppies to begin with.

While this can be taken care of using rules, it still leaves some room for error because it is important to change the equation as well. And in cases where the quantities in the word problem are same, the task doesn't seem all that trivial.

- **Synonym Replacement:** As demonstrated by [32], common nouns can be replaced by their synonyms in the word problem. For example:

  - **Original:** Alyssa 's dog had puppies . She gave 7 to her friends . She now has 5 puppies . How many puppies did she have to start with ?

    After Substitution: Alyssa 's cat had kittens. She gave 7 to her parents. She now has 5 kittens. How many kittens did she have to start with?

  - **Original:** There are 2 pencils in the drawer . Tim placed 3 pencils in the drawer . How many pencils are now there in total ?

    After Substitution: There are 2 crayons in the cupboard . Tim placed 3 crayons in the cupboard . How many crayons are now there in total ?

3.1.1.2 Paraphrasing

As the name suggests, in this method the word problems in the dataset are paraphrased. This preserves the semantic and syntactic meaning while also preserving the equations and other annotations. Examples are given below.

- **Original:** Alyssa's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with?

  After paraphrasing: The dog Alyssa owns gave birth to pups. She handed 7 of them to her pals and is now raising 5 pups. How many pups did she have at the beginning?

3.1.1.3 Reverse Operation

This method converts the original question statement into a assertive statement with a definite quantity and converts the statement with the definite quantity into a question. Examples:

- **Original:** Alyssa's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with?

  After Reverse Operation: Alyssa's dog had 12 puppies. She gave some to her friends. She now has 5 puppies. How many puppies did she give to her friends?

3.1.1.4 Back Translation

Also known as round-trip translations, this approach gives a paraphrase of the original using a set of translations. The method involves translating sentences into foreign languages, then translating them back into the original language. There are can multiple intermediary translations between multiple languages as well. Examples of word problems augmented using back translation (produced from Machine Translation systems) include:

- **Original: (English)** Alyssa's dog had puppies. She gave 7 to her friends. She now has 5 puppies. How many puppies did she have to start with?

  After Translation 1: (Hindi) elisa ke kutte ke pille the. usane apane doston ko 7 die. usake ab 5 pille hain. use kitane pillon se shuruaat karanee thee?

  After Translation 2 (Spanish): El perro de Alyssa tuvo cachorros. Le dio 7 a sus amigos. Ahora tiene 5 cachorros. ¿Con cuántos cachorros tuvo al principio?

  After Translation 3 (English): Alyssa's dog had puppies. He gave 7 to her friends. He now has 5 puppies. How many pups did he have at first?

3.1.2 Data Augmentation Techniques used for HAWP

Most augmentation techniques mentioned above have a pain point - they do not produce diverse data. Because of this, the lexical diversity as well as the syntactic diversity of the dataset, decreases. Even if we begin with a highly diverse dataset, during the augmentation process, we generate near-same copies of each sample, thus lowering the diversity of the dataset. Because of this, the models learn a limited number of equation types and similar text patterns, failing to perform well on unseen data.

To avoid such problems, we used translations of already existing benchmark datasets as a data augmentation strategy. The following section is dedicated to this technique. We discuss the approach and also explain how it was used such that we avoid facing the problems faced with other methods.

3.2 Augmentation using Translation

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.

3.2.1 English datasets used for translation

We used AI2 [10], Unbiased [27], ASDiv [20] datasets for this task. The motivation behind choosing these datasets is that they 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.

3.2.2 Translation of English MWPs

Professional translators performed the task of translation, and it was done in two stages.

A batch of randomly selected English word problems was manually translated to Hindi. While doing so, it was realised that this was a very expensive approach. We used the translation tools available to simplify the process.

Before we moved on to using machine translators for the entire process, we performed machine translation(MT) on the same batch of word problems that were manually translated. We compared the two translations to understand whether this was a viable approach, the issues we would face and whether solving those issues could be more expensive than translating the word problems manually.

Issues were found, as expected. We took some time to identify patterns in these issues and categorise them to make post-editing easier. Additionally, we had to keep in mind that these translated word problems should sound just as natural as Hindi word problems. And to find and correct the issues in translations, a post-edit tool was used where translations from multiple MT tools were provided for reference.

With the understanding of issues, patterns/categories serving as guidelines for dealing with them and with the added ease of using the post-edit tool, we decided to translate the rest of the problems using machine translation tools to minimise overall time and effort.

Correction of these issues, along with other challenges are explained in the following subsections.

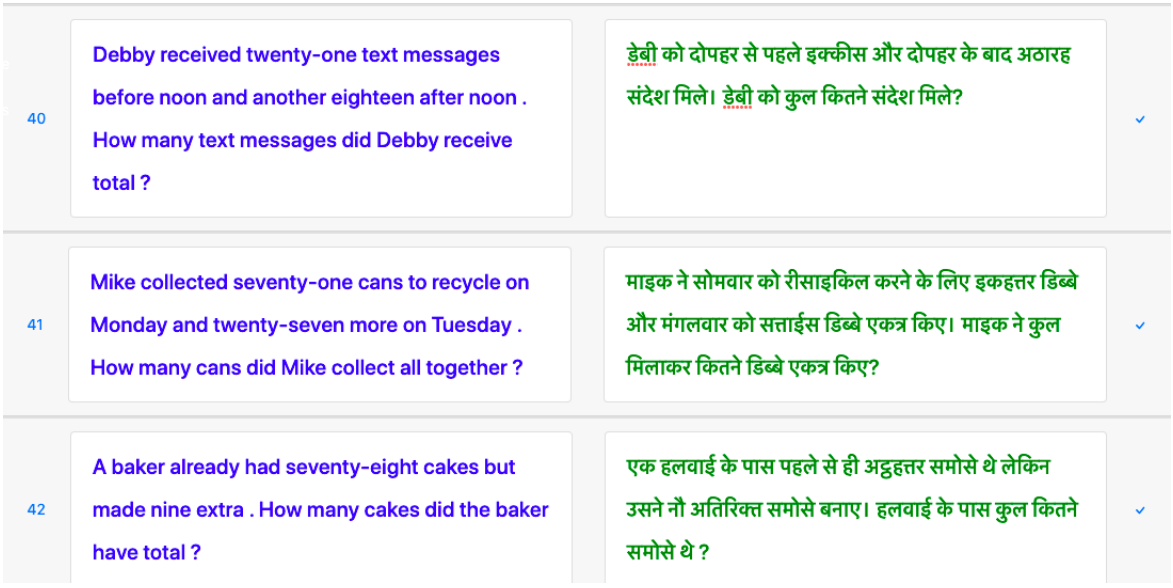


Figure 3.1: Snapshot of the Post Editing Tool

3.2.3 Challenges in data augmentation using translation

After studying the translated word problems, we grouped the issues found in those translations into categories. While some issues were more of linguistic errors, some were not really errors but more of linguistic or cultural mismatches. All these errors and mismatches are discussed below in detail with examples:

3.2.3.1 Errors encountered in translated data

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 3.1.

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

Table 3.1: Types of errors encountered while augmenting Hindi word problems using Machine Translation.

As shown in Table 3.1, each error category has a sub-category. For example, PP attachment is a syntactic error; hence it belongs to that category. Similarly, discourse errors noted above may contain inconsistencies in translation within the discourse of the same MWP. We discuss examples from different categories below to get a deeper insight into these errors and the patterns in which they appear in the Machine Translated output. Each example has 4 components:

- MWP in English - Input to MT system
- MWP in Hindi (Devanagari) - Output given by MT system in Devanagari script
- MWP in Hindi (Roman) - Output given by MT system in Roman script
- Description of error in the example.

1. **Wrong Tense/Aspect/Mood (TAM)**

(a) 'Kelly has 121.0 Nintendo games. How many does Kelly need to give away so that Kelly will have 22.0 games left?'

केली के पास 121.0 निन्टेंडो गेम्स हैं। केली को कितने देने होंगे ताकि केली के पास 22.0 गेम बचे हों?

kelee ke paas 121.0 nintendo gems hain. kelee ko kitane dene honge taaki kelee ke paas 22.0 gem bache hon?

Error: Wrong TAM used in verb 'bach' as 'bache hon' instead of 'bache'.

(b) '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?'

आपकी कक्षा में पिज़्ज़ा पार्टी थी। एक पिज्जा का 0.375 बचा हुआ था, और दूसरे पिज्जा का 0.5 बचा हुआ था। आप उन दोनों को 1.0 बॉक्स में डाल दें। आपके पास कुल मिलाकर कितना पिज्जा है?

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?

Error: There are cases in which TAM may change depending upon the discourse. In this example, the translated mood is imperative. However, given the context of the MWP, the mood should be indicative.

2. **PP Attachment**

(a) 'Sara picked 45 pears and Sally picked 11 pears from the pear tree. How many pears were picked in total?'

सारा ने 45 नाशपाती और सैली ने नाशपाती के पेड़ से 11 नाशपाती लिए। कुल कितने नाशपाती चुने गए?

saara ne 45 naashapaatee aur sailee ne naashapaatee ke ped se 11 naashapaatee lie. kul kitane naashapaatee chune gae?

Error: The prepositional phrase in the translated problem i.e. from the pear tree gets attached to only one of the verb phrases i.e Sally picked 11 pears and misses the other i.e. Sara picked 45 pears.

(b) 'Faye had 46.0 math problems and 9.0 science problems for homework. If she finished 40.0 of the problems at school, how many problems did she have to do for homework?'

फेय में 46.0 गणित और होमवर्क के लिए 9.0 विज्ञान की समस्याएं थीं। अगर उसने स्कूल में 40.0 समस्याएं पूरी कीं, तो होमवर्क के लिए उसे कितनी समस्याएं उठानी पड़ीं?

phey mein 46.0 ganit aur homavark ke lie 9.0 vigyaan kee samasyaen theen. agar usane skool mein 40.0 samasyaen pooree keen, to homavark ke lie use kitanee samasyaen uthaanee padeen?

Error: The PP in the translated problem i.e. for homework gets attached to only one of the noun phrases i.e 9 science problems and misses the other i.e. 46 math problems.

3. **Incorrect Number/Missing Pluralisation**

(a) 'Carol and her mom were picking carrots from their garden. Carol picked 29.0 , and her mother picked 16.0. If they picked another 38.0, how many bad carrots did they have?'

कैरल और उसकी माँ अपने बगीचे से गाजर उठा रहे थे। कैरल ने 29.0 चुना, और उसकी मां ने 16.0 उठाया। अगर उन्होंने एक और 38.0 उठाया, तो उनके पास कितनी खराब गाजर थी?

kairal aur usakee maan apane bageeche se gaajar utha rahe the. kairal ne 29.0 chuna, aur usakee maan ne 16.0 uthaaya. agar unhonne ek aur 38.0 uthaaya, to unake paas kitanee kharaab gaajar thee?

Error: In Hindi, the verb carries the Number information as inflection. In this case, verbs 'uthaya' or 'chuna' (picked) are not inflected for pluralisation.

(b) 'Shannon, Brenda's neighbor, joined Brenda in making bracelets. She brought 48.0 heart-shaped stones and wanted to have 8.0 of this type of stone in each of the bracelet she makes. How many bracelets with heart-shaped stones can Shannon make?'

ब्रेंडा के पड़ोसी शैनन ब्रेंडा के साथ ब्रेसलेट बनाने में शामिल हुए। वह 48.0 दिल के आकार के पत्थर लाई और चाहती थी कि उसके द्वारा बनाए गए प्रत्येक ब्रेसलेट में इस प्रकार का 8.0 पत्थर हो। शैनन दिल के आकार के पत्थरों से कितने कंगन बना सकता है?

brenda ke padosee shainan brenda ke saath bresalet banaane mein shaamil hue. vah 48.0 dil ke aakaar ke patthar laee aur chaahatee thee ki usake dvaara banae gae pratyek bresalet mein is prakaar ka 8.0 patthar ho. shainan dil ke aakaar ke pattharon se kitane kangan bana sakata hai?

Error: Missing inflection. In this case, postposition or case marker 'ka' (of) is not inflected for pluralisation.

4. **Incorrect Postpositions/Case Marking**

(a) 'Jason joined his school's band. He bought a flute for $142.46, a music stand for $8.89, and a song book for $7.0. How much did Jason spend at the music store?'

जेसन अपने स्कूल के बैंड में शामिल हो गया। उन्होंने ₹142.46 के लिए एक बांसुरी, ₹8.89 के लिए एक संगीत स्टैंड और ₹7.0 के लिए एक गीत पुस्तक खरीदी। जेसन ने संगीत की दुकान पर कितना खर्च किया?

jesan apane skool ke baind mein shaamil ho gaya. unhonne ₹142.46 ke lie ek baansuree, ₹8.89 ke lie ek sangeet staind aur ₹7.0 ke lie ek geet pustak khareedee. jesan ne sangeet kee dukaan par kitana kharch kiya?

Error: The word 'for' is mistranslated to the wrong post position 'ke liye' as opposed to 'ki'.

(b) 'Jane helped her mom prepare fresh lemonade. If each glass needs 2.0 lemons, how many glasses of fresh lemonade can she make if they have 18.0 lemons?'

जेन ने अपनी माँ को ताज़ा नींबू पानी बनाने में मदद की। यदि प्रत्येक गिलास में 2.0 नींबू की आवश्यकता है, तो वह 18.0 नींबू होने पर कितने गिलास ताजा नींबू पानी बना सकती है?

jen ne apanee maan ko taaza neemboo paanee banaane mein madad kee. yadi pratyek gilaas mein 2.0 neemboo kee aavashyakata hai, to vah 18.0 neemboo hone par kitane gilaas taaja neemboo paanee bana sakatee hai?

Error: Wrong post position 'ko' as opposed to 'ki' is used.

(c) 'A company invited 18.0 people to a luncheon, but 12.0 of them didn't show up. If the tables they had, held 3.0 people each, how many tables do they need?'

एक कंपनी ने 18.0 लोगों को लंच पर आमंत्रित किया, लेकिन उनमें से 12.0 लोग नहीं आए। यदि उनके पास जो टेबल थी, उनमें प्रत्येक में 3.0 लोग थे, तो उन्हें कितनी टेबल की आवश्यकता होगी?

ek kampanee ne 18.0 logon ko lanch par aamantrit kiya, lekin unamen se 12.0 log nahin aae. yadi unake paas jo tebal thee, unamen pratyek mein 3.0 log the, to unhen kitanee tebal kee aavashyakata hogee?

Error: The correct use is 'pratyek table par 3 log the' and not

5. **Wrong Sense**

(a) '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?'

ल्यूक अपने अतिरिक्त परिवर्तन को बवासीर में डाल रहा था। उसके पास 5.0 पाइल्स ऑफ़ क्वार्टर्स और 5.0 पाइल्स ऑफ़ डाइम्स थे। यदि प्रत्येक ढेर में 3.0 सिक्के हों, तो उसके पास कुल कितने सिक्के थे?

lyook apane atirikt parivartan ko bavaaseer mein daal raha tha. usake paas 5.0 pails of kvaar-tars aur 5.0 pails of daims the. yadi pratyek dher mein 3.0 sikke hon, to usake paas kul kitane sikke the?

Error: Wrong senses of change (money related change vs. alteration/modification) and piles (plural of pile (chunk/hill) vs. piles(medical consition)) are used.

(b) '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?'

एक दराज में 14.0 शासक और 34.0 क्रेयॉन होते हैं। टिम 11.0 शासकों को दराज से निकालता है। दराज में अब कितने शासक हैं?

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?

Error: Wrong sense of ruler - scale vs. emperor.

6. **Inconsistent Use of Honourifics**

(a) 'Dan found 56.0 oysters on the beach, he gave some of his seashells to Jessica. He has 22.0 oysters. How many shells did he give to Jessica?'

डैन को समुद्र तट पर 56.0 सीप मिले, उन्होंने जेसिका को अपने कुछ सीशेल्स दिए। उसके पास 22.0 सीपियां हैं। उसने जेसिका को कितने सीपियाँ दीं?

dain ko samudr tat par 56.0 seep mile, unhonne jesika ko apane kuchh seeshels die. usake paas 22.0 seepiyaan hain. usane jesika ko kitane seepiyaan deen?

Error: Here, honorific pronoun ('unhonne') is used in first statement while in others general pronoun ('usane', 'usko') is used.

(b) 'Adam had 5.0 dollars. At the store he spent $2.0 on a new game. If he got another 5.0 dollars for his allowance, how much money does he have now?'

एडम के पास 5.0 डॉलर थे। स्टोर पर उन्होंने एक नए गेम पर ₹2.0 खर्च किए। यदि उसे अपने भत्ते के लिए और 5.0 डॉलर मिलते हैं, तो उसके पास अब कितना पैसा है?

edam ke paas 5.0 dolar the. stor par unhonne ek nae gem par ₹2.0 kharch kie. yadi use apane bhatte ke lie aur 5.0 dolar milate hain, to usake paas ab kitana paisa hai?

Error: Here, honorific pronoun ('unhonne') is used in first statement while in others general pronoun ('use', 'usake') is used.

7. **Incorrect Translation of Phrasal Verbs**

(a) 'A trivia team had 5.0 members total, but during a game 2.0 members didn't show up. If each member that did show up scored 6.0 points, how many points were scored total? '

एक सामान्य ज्ञान टीम में कुल 5.0 सदस्य थे, लेकिन एक गेम के दौरान 2.0 सदस्य नहीं दिखाई दिए। यदि प्रदर्शन करने वाले प्रत्येक सदस्य ने 6.0 अंक अर्जित किए, तो कुल कितने अंक प्राप्त हुए?

ek saamaany gyaan teem mein kul 5.0 sadasy the, lekin ek gem ke dauraan 2.0 sadasy nahin dikhaee die. yadi pradarshan karane vaale pratyek sadasy ne 6.0 ank arjit kie, to kul kitane ank praapt hue?

Here the meaning of show up is incorrectly translated to senses 'seen' and 'display'.

8. **Wrong Numeral**

(a) 'Olivia gave her cat two cheese cubes. Now Olivia has ninety-eight cheese cubes left. How many cheese cubes did Olivia have originally?'

ओलिविया ने अपनी बिल्ली को पनीर के दो टुकड़े दिए। अब ओलिविया के पास निन्यानबे पनीर क्यूब्स बचे हैं। ओलिविया के पास मूल रूप से कितने पनीर क्यूब्स थे?

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?

Error: 98 is 'atthaanabe' while 'ninyaanabe' is 99

(b) 'Henry earned eighty nine dollars for each lawn he mowed. If he had twelve lawns to mow, but forgot to mow seven of them, how much money did he actually earn? '

हेनरी ने अपने द्वारा काटे गए प्रत्येक लॉन के लिए अस्सी नौ डॉलर कमाए। यदि उसके पास घास काटने के लिए बारह लॉन थे, लेकिन उनमें से सात को काटना भूल गए, तो उसने वास्तव में कितना पैसा कमाया?

henaree ne apane dvaara kaate gae pratyek lon ke lie assee nau dolar kamae. yadi usake paas ghaas kaatane ke lie baarah lon the, lekin unamen se saat ko kaatana bhool gae, to usane vaastav mein kitana paisa kamaaya?

Error: Here number terms are written without hyphen due to which they are considered as two separate numbers: eighty and nine and thus get translated as 'assee'(eighty) and 'nau'(nine).

Note: These issues were found when we started the exercise of using Machine Translations for augmentation. MT systems have improved now and not all of these errors may be reproducible.

3.2.3.2 Missing one-to-one mapping.

Even though we embodied localisation and borrowing 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. This is the issue relating to the "linguistic mismatch" mentioned earlier. While this is more of a translation issue, this affects the translated word problems' naturalness and sometimes makes the word problem wrong. Examples include:

- The two different English concepts, 'running' and 'sprinting' get translated to the same 'daudna' in Hindi. Similarly, 'road' and 'street' are loosely translated to 'sadak' in Hindi. An example of how this issue affects the translated word problems:

  English: Darnel sprinted 0.98 lap and then took a break by running 0.50 lap. How much farther did Darnel sprint than run?

  Machine Translated Hindi: डेरनेल ने 0.98 लैप दौड़ लगाई और फिर 0.50 लैप दौड़कर ब्रेक लिया। डैरनेल ने दौड़ने की तुलना में कितनी दूर तक दौड़ लगाई?

  deranel ne 0.98 laip daud lagaee aur phir 0.50 laip daudakar brek liya. dairanel ne daudane kee tulana mein kitanee door tak daud lagaee?

  Issue: Here the solver would not understand the word problem as both activities being compared in the question are same.

- On the other hand the same verb 'serve' are translated to 'parosna', 'daalna', 'dena' depending on the recipient. An example of how this issue affects the translated word problems:

  English: When Jake had 1.0 cat, he needed to serve 0.5 can of cat food each day. Now that Jake has adopted a second cat, he needs to serve a total of 0.9 can each day. How much extra food is needed to feed the second cat?

  Machine Translated Hindi: जब जेक के पास 1.0 बिल्ली थी, तो उसे प्रतिदिन 0.5 कैन बिल्ली का भोजन परोसने की आवश्यकता होती थी। अब जब जेक ने दूसरी बिल्ली गोद ले ली है, तो उसे हर दिन कुल 0.9 कैन परोसने की ज़रूरत है। दूसरी बिल्ली को खिलाने के लिए कितना अतिरिक्त भोजन चाहिए?

jab jek ke paas 1.0 billee thee, to use pratidin 0.5 kain billee ka bhojan parosane kee aavashyakata hotee thee. ab jab jek ne doosaree billee god le lee hai, to use har din kul 0.9 kain parosane kee zaroorat hai. doosaree billee ko khilaane ke lie kitana atirikt bhojan chaahie?

Issue: For animals, using 'parosna' for the verb 'serve' is not natural in Hindi. This impacts the naturalness of the word problem.

3.2.3.3 Cultural differences.

Since we took benchmark datasets in English, we noticed cultural differences in word problems in the kind of objects, events, and names used, making the translated Hindi word problems less natural.

These issues were very carefully handled with the help of some pointers/guidelines that were kept in mind while post-editing. In the following part of this section, we dive into what these pointers were and with a few examples, we see how the translated Hindi MWPs were improved.

3.2.4 Correction of issues seen in translated MWPs

To handle the problems in the translated data and ensure that the dataset does not stray away from the typical nature that word problems follow and, at the same time, has Hindi fluency and grammatical correctness, we laid down some key points to keep in mind for both human translation and post-editing. These points were circulated to the post-editors with sufficient examples. These can be used as a standard guideline for future MWP data augmentation for ILs:

3.2.4.1 Correction of errors in translated MWPs

Firstly, the errors identified after translation were corrected to use the correct TAM, PP Attachment, case markings etc. Missing words were added while incorrect/irrelevant ones were removed or replaced with correct words.

Table 3.2 shows examples of how the errors encountered while augmenting Hindi word problems using Machine Translation were corrected.

Type of Error	Example		
	English	Machine Translated	After Post-Editing
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 naasha-paatee aur sailee ne naashapaatee ke ped se 11 naashapaatee lie. kul kitane naashapaatee chune gae?	naashapaatee ke ped se saara ne 45 naashapaatee aur sailee ne 11 naasha-paatee lie. kul kitane naashapaatee chune gae?

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?	anaaj se bhara 1 jahaaz koral reef mein durghatanaagrast ho gaya. jab tak jahaaz kee marammat hotee, tab tak 49952 tan anaaj paanee mein phail chuka tha. jahaaz par keval 918 tan anaaj bacha. to jahaaz par moolat: kitane tan anaaj tha?
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?	keya ke paas 121 nintendo gem hain. keya ko kitane dene kee zaroorat hai taaki keya ke paas 22 gem bachen?
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?	aapakee kaksha mein pizza paartee thee. 1 pizza ka 0.375 hissa bacha hua tha aur **doosare pizza ka 0.5 hissa** bacha hua tha. aapane un donon ko 1 dibbe mein rakh diya. aapake paas kul milaakar kitana pizza hai?
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?	skool kaifeteriya ne chhaatron ko dopahar ke khaane mein 42 laal seb aur 7 hare seb die. lekin, agar keval 9 chhaatron ko phal chaahie the, to kainteen mein kitane phal atirikt rah gae?

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?	lyook apane chhutte paison ke dher bana raha tha. usake paas chavannee ke 5 aur athannee ke 5 dher the. yadi pratyek dher mein 3 sikke the, to usake paas kul kitane sikke the?
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?	ek daraaj mein 14.0 scale aur 34.0 kreyon the. tarun ne 11.0 scale daraaj se nikaale. daraaj mein ab kitane scale hain?
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?	daivik ko samudr tat par 56 seepiyaan mileen, usane laila ko apanee kuchh seepiyaan deen. usake paas ab 22 seepiyaan hain. usane laila ko kitanee seepiyaan deen?
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?	durga ne apanee billee ko paneer ke 2 tukade die. ab durga ke paas 98 paneer ke tukade bache hain. durga ke paas moolat: paneer ke kitane tukade the?
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?	firoz ke paas 26 chikan wings the aur usne mairee ko 18 wings die. phir use 40 ka ek khula hua dibba mila. usake paas kul kitane chikan wings hain?

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?	jyoti ne 2 rang bharane kee kitaaben khareedeen. 1 mein 23 chitr the aur doosare mein 32 the. 1 saptaah ke baad, vah 44 chitron mein rang bhar chukee thee. use abhee bhee kitane cheeton mein rang bharana hai?

Table 3.2: Examples of Translation Errors

3.2.4.2 Localisation

We used localisation for the translated MWPs to correctly adapt to India or more specifically, to the Hindi language. A group of native Hindi speakers carried out this process. 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 3.3.

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 3.3: Examples of Localisation

3.2.4.3 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 3.4

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 3.4: Examples of Borrowing

3.2.4.4 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:

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

Gloss: Bina has 63 sweets. I have 50 sweets. How many sweets do we have?

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

After: bina ke paas 63 mithaiyaan hain. mere paas 50 mithaiyaan hain. hamaare paas **kul** kitanee mithaiyaan hain?

Gloss: Bina has 63 sweets. I have 50 sweets. How many sweets do we have **altogether**?

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 3.5.

3.2.4.5 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

English:	Molly had 14 candles on her birthday cake. **She grew older** and *got 6 more* on her birthday cake. How old is Molly now?
Direct Translation:	maulee ke barthade kek par 14 momabattiyaan theen. **vah badee ho gaee** aur usake barthade kek par use *6 aur mileen*. maulee ab kitanee badee hai?
After Post Editing:	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?
Remarks:	More details have been added to the problem and some parts have been changed to make it more natural.
English:	Mrs. Sheridan has 22.0 fish. Her sister gave her 47.0 more fish. **How many fish does she have now?**
Direct Translation:	shreematee lata ke paas 22 machhaliyaan hain. unakee bahan ne unhen 47 machhaliyaan aur deen. **ab unake paas kitanee machhaliyaan hain?**
After Post Editing:	shreematee lata ke paas 22 machhaliyaan theen. unakee bahan ne unhen 47 machhaliyaan aur deen. **ab unake paas *kul* kitanee machhaliyaan hain?**
Remarks:	”kul” has been added to make the word problem more natural.

Table 3.5: Examples of Naturalness for Augmentation using Translation

crowdsourcing. The identified mistakes were also corrected as part of post editing. Table 3.6 shows some examples and how and why they were corrected.

3.2.4.6 Diversity of MWPs

When it comes to the diversity of a dataset, the more, the better. The benchmark datasets used for augmentation have different types of MWPs. Unbiased dataset [27] has many MWPs that have introduced irrelevant information, while ASDiv [20] has MWPs with very high lexical diversity.

Other than the presence of irrelevant information and lexical diversity, researchers have shown that there are different ways to increase the quality of a dataset. [23] have stressed the importance of having challenging problems that pose a real test on solvers' attention and reasoning ability. 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 to these variants of the same problem, the solver must pay attention to even the smallest change in the word problem and 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 are the 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).

English:	Mrs. Sheridan **has** 22.0 fish. Her sister gave her 47.0 more fish. How many fish does she have now?
Direct Translation:	shreematee sheridan ke paas 22 machhaliyaan **hain**. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas kitanee machhaliyaan hain?
After Post Editing:	shreematee lata ke paas 22 machhaliyaan **theen**. unakee bahan ne unhen 47 machhaliyaan aur deen. ab unake paas kul kitanee machhaliyaan hain?
Remarks:	Past tense should be used to describe the state before a change or transaction instead of present tense.
English:	Isabella's hair is 18.0 inches long. **If her hair grows 4.0 more inches, how long will it be?**
Direct Translation:	izaabel ke baal 18 inch lambe the. **yadi usake baal 4 inch badhe, ve kitane lambe honge?**
After Post Editing:	gauree ke baal 18 inch lambe the. **yadi usake baal 4 inch badhe, *to* usake baal kitane lambe honge?**
Remarks:	Conditional sentences using 'yadi'(if) require 'to'(then) in Hindi.

Table 3.6: Examples of Grammatical Correctness for Augmentation using Translation

Implicit:	faatima ke bageeche mein 8 aaloo the. **kharagoshon ne 3 kha lie.** faatima ke paas ab kitane aaloo hain? Fatima had 8 potatoes in her garden. **The rabbits ate 3.** How many potatoes does Fatima have now?
Explicit:	faatima ke bageeche mein 8 aaloo the. **kharagoshon ne un aalooon mein se 3 kha lie.** faatima ke paas ab kitane aaloo hain? Fatima had 8 potatoes in her garden. **The rabbits ate 3 *of the potatoes*.** How many potatoes does Fatima have now?

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 3.7. These changes may or may not change the answer to the word problem.

With the help of data augmentation, we were able to achieve a dataset of 2336 Hindi MWPs. Before using these word problems to build solvers, we evaluate our dataset on the properties of naturalness and diversity in the next chapter.

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*?** Gloss: Ram went to watch 11 cricket matches this month. He went to watch 17 matches last month and next month he will go to watch 16 matches. **How many matches *will he watch?***	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?*** Gloss: Ram went to watch 11 cricket matches this month. He went to watch 17 matches last month and next month he will go to watch 16 matches. **How many matches *has he watched?***	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*?** Gloss: Ram went to watch 11 cricket matches this month. He went to watch 17 matches last month and next month he will go to watch 16 matches during the day and 12 matches at night. **How many matches will he watch?**	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? Gloss: Rani works at a sweet shop. She sold 45 on Monday. **She sold 16 on Tuesday.** How many sweets did she sell?	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? Gloss: Rani works at a sweet shop. She sold 45 on Monday. **On Tuesday she sold 16 less.** How many sweets did she sell?	X = 45+(45-16)	Added quantifier

Table 3.7: Variant MWPs to Increase Diversity of Problems

Chapter 4

Nature of Dataset

The HAWP dataset has the following properties:

- Natural: The word problems are natural for school children for solving
- Diverse: The dataset has a rich lexical diversity

4.1 Naturalness

The naturalness of the dataset is governed by the solvability of word problems 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 Grades 6-7. Problems were picked randomly and grouped into batches of 15. Each student was instructed to solve one batch of problems on paper. Therefore, a total of 75 unique problems were evaluated in this process.

Most students wrote the answer of the problem directly while two students wrote detailed steps showing how they reached the solution to each problem. Overall, 89.33% of the MWPs (i.e. 67 out of 75 word problems) were solved with correct answers. The students who wrote detailed solutions could form correct equations for 90% of the MWPs (i.e. 27 out of 30 word problems).

Among the incorrectly solved word problems, almost 85% (i.e. 7 out of 8 incorrectly solved MWPs) 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 of the quality of localisation, borrowing and naturalness of the dataset, and HAWP cleared this challenge.

4.2 Diversity of word problems in the dataset

To measure the degree of diversity of problems in HAWP, we used several diversity metrics proposed by different researchers. For metrics dealing with lexical diversity, we understand that different languages can have different lexical overlaps. Hence the scores for English datasets have been listed only for reference.

4.2.1 MAWPS Lexical Diversity

We calculated the lexical overlap of HAWP as proposed by [15] 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**.

Dataset	Lexical Diversity
MAWPS (for single equation problems)	6.52%
ASDiv (complete dataset)	5.84%
HAWP (complete dataset)	5.92%

Table 4.1: Lexical Diversity for different datasets

4.2.2 Corpus Lexicon Diversity(CLD)

We also found the corpus lexicon usage diversity metric, CLD as proposed by [20]. 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 [22] 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**.

Dataset	Lexical Diversity
MAWPS (for single equation problems)	0.42
ASDiv (complete dataset)	0.49
HAWP (complete dataset)	0.73

Table 4.2: CLD for different datasets

4.2.3 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 4.3 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 4.3: Reduction of Datasizes after Removal of Similar MWPs

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

The work presented till now is the first step towards building Hindi Word Problem Solvers. Now that our dataset is well prepared, we will use this dataset to develop solvers that can generate equations given a word problem. In the following two chapters, we will look at two methods that can be used to solve Hindi word problems.

Chapter 5

Hindi Word Problem Solver: Verb Categorisation for Word Problem Solving

This chapter consists of one of the two Hindi MWP solvers discussed as part of this thesis. Our first solver uses verb categorisation to identify operations and generate answers for the word problem. To perform verb categorisation, we explored several approaches and listed their pros and cons. The motivation behind using verb categorisation for Word Problem Solving was to exploit the semantics in MWPs to extract equations from them.

5.1 Introduction

If we look closely, it is the semantics of the world problem that eventually reduces to simple equations. Hence, many word problem solvers solve the MWPs by parsing the semantics of the word problems and generating a different representation of this information. This way, the word problems are solved in two major steps:

- **Step 1:** Converting the given word problem to a logical form that captures the meaning of the word problem to extract the precise and focused meaning that will facilitate identifying the operations involved in the word problem.
- **Step 2:** Using this logical form to find the equation associated with the word problem.

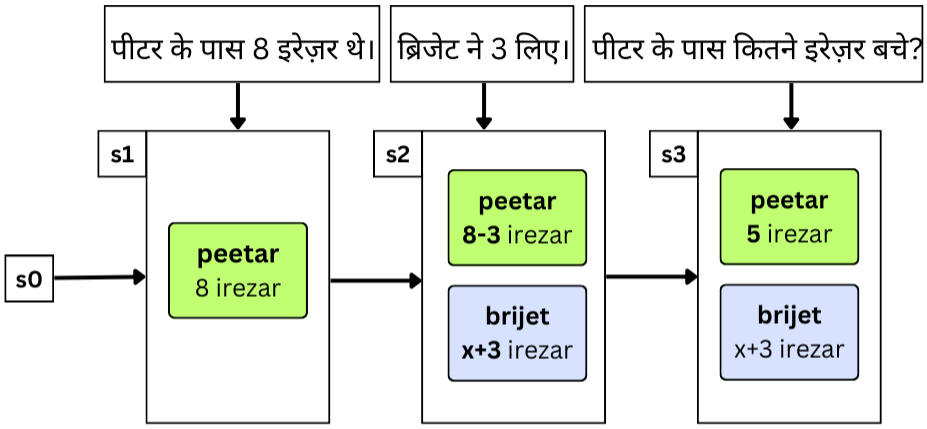
Verb Categorisation is one such approach. Among the many semantic parsing approaches to word problem solving, this appears to be the most intuitive one.

This approach, introduced in [10], looks at each arithmetic word problem as text describing a partial world state, which changes by simple updates and ends with a question regarding any of the states: initial, intermediate or final state.

State Representations: Each word problem is viewed as a sequence of states. Each state gives us a snapshot of the situation at a time in the word problem - 'who' has 'how many' of 'what' in a given statement in the word problem. A state consists of the following pieces of information:

- **Entities**: Objects whose quantity is observed or updated through the course of the word problem. 'Irrezar' (erasers) is the entity in our example.
- **Attributes of Entities**: A characteristic quality or feature of an entity. These are usually marked by adjectives, example: 'pink' in pink erasers.
- **Containers**: A container refers to a group of entities. It may refer to any animate/inanimate object that possesses or contains entities. 'peetar' (Peter) and 'brijet' (Bridget) and is the entity in our example.
- **Quantities of Entities**: The number of entities in a given state. Quantities in a state can be known or unknown. 8 and 3 state the quantities in our example.

This can be understood using the following example:

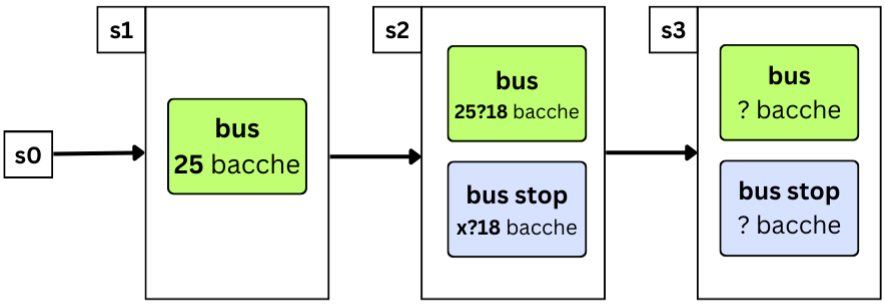


पीटर के पास 8 इरेज़र थे। ब्रिजेट ने 3 लिए। पीटर के पास कितने इरेज़र बचे?
peetar ke paas 8 irezar the. brijet ne 3 lie. peetar ke paas kitane irezar bache?
Gloss: Peter had 8 erasers. Bridget took 3. How many erasers are left with Peter?

Figure 5.1: Word Problem as State Transitions

Therefore, verb categorisation maps the word problems to a **state representation**. To make transitions from one state to another (or one statement to another), the verb associated with that state is used. To understand how this happens, first, let us take an example and understand the vital role of verbs in solving word problems. Given the figure below can you identify the answer to the question?

बस में 25 बच्चे सवार थे। बस स्टॉप आने पर 18 बच्चे <>। अब कितने बच्चे बस में सवार हैं?

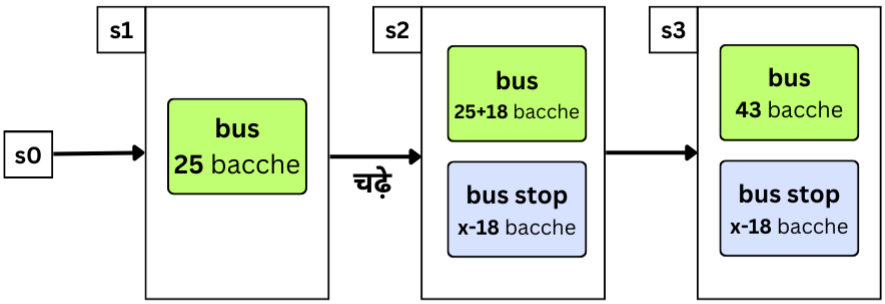


bas mein 25 bachche savaar the. bas stop aane par 18 bachche <>. ab kitane bachche bas mein savaar hain?
Gloss: There were 25 children on the bus. 18 children <> at the bus stop. How many children are on the bus now?

Figure 5.2: State Transitions without verbs are difficult

Given states with containers and entities and their quantities but no verbs, we are not be able to identify the operation that needs to be done to reach the final state and answer the question asked in the word problem.

बस में 25 बच्चे सवार थे। बस स्टॉप आने पर 18 बच्चे चढ़े। अब कितने बच्चे बस में सवार हैं?

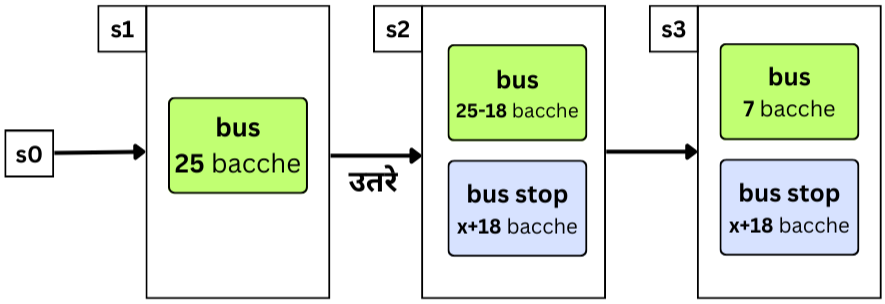


bas mein 25 bachche savaar the. bas stop aane par 18 bachche chadhe. ab kitane bachche bas mein savaar hain?
Gloss: There were 25 children on the bus. 18 children got on at the bus stop. How many children are on the bus now?

Figure 5.3: State Transitions with verbs

As we change the verb, the operation also changes:

बस में 25 बच्चे सवार थे। बस स्टॉप आने पर 18 बच्चे उतरे। अब कितने बच्चे बस में सवार हैं?



bas mein 25 bachche savaar the. bas stop aane par 18 bachche utare. ab kitane bachche bas mein savaar hain?
Gloss: There were 25 children on the bus. 18 children got off at the bus stop. How many children are on the bus now?

Figure 5.4: State Transitions with verbs: Changing the verb, changes the operation

The verbs give us the information about the operation to be performed and hence are a crucial part of solving word problems, which is why the focus is on verbs! Next up, let us understand how verbs can be used to identify mathematical operations.

5.2 Verb Categorisation

This section is focused on the first step of word problem solving using verb categorisation, i.e. classifying verbs into semantic categories. This stage on its own has a number of sub-steps, starting from identifying the number and type of categories suitable for word problems, annotating the verbs in the dataset with these categories and finally, building and using statistical models to predict the category of any verb. Hence, this is a small problem statement on its own, for which we first need to create a dataset and then a model.

Since verbs can be used to identify only positive and negative operations, we filtered the HAWP dataset to have only word problems involving addition and subtraction operations. For multiplication and division we will need to categorise constituents other than verbs as well.

5.2.1 Verb Categories

If we look at verb categorisation as a machine learning problem, verb categories serve as the labels - the thing we are predicting. [10] identifies seven verb categories, analysing the "genre" of MWPs in their dataset, AI2.

Verb Category	**Definition**
Observation	the quantity is initialized in the container.
Positive	the quantity is incremented in the container.
Negative	the quantity is decreased in the container.
Positive Transfer	the quantity is moved from the second container to the first one.
Negative Transfer	The quantity is moved from the first container to the second one.
Construct	The quantity is incremented for both containers.
Destroy	The quantity is decreased for both containers.

Table 5.1: Original Seven Verb Categories

Inspired by these seven, we created the following five categories:

1. **Observation:** It states just the presence of entities in a container. It has no operation on its own. It is not restricted to initialisation (in initial state) but may also imply assignment (in any state) of a quantity in a container. Examples:
   - दराज में 9 क्रेयॉन थे।

     daraaj mein 9 kreyon the.

     Gloss: There were 9 crayons in the drawer.
   - मेलिसा के पास 88 केले थे।

     melisa ke paas 88 kele bache.

     Gloss: Melissa has 88 bananas left.
2. **Positive:** It states the quantity of entities being added to a container or which are created in a container. It indicates the operation of addition. Examples:
   - मरीला अस्पताल में थी और उसे देश भर से 403 शुभकामना कार्ड मिले।

     mareela aspataal mein thee aur use desh bhar se 403 shubhakaamana kaard mile.

     Gloss: Mariela was in the hospital and received 403 greeting cards from across the country.
   - विदूषक ने 47 गुब्बारे फुलाए।

vidooshak ne 47 gubbaare phulae.

Gloss: The clown inflated 47 balloons.

3. **Negative:** It states the quantity of entities being removed or destroyed from a container. It indicates the operation of subtraction. Examples:

   - 4 सीपियाँ टूट गईं।

     4 seepiyaan toot gaeen.

     Gloss: 4 seashells were broken.

   - 2 आइसक्रीम पिघलकर ख़राब हो गई।

     2 aaisakreem pighalakar kharaab ho gaee.

     Gloss: 2 ice creams melted and got spoiled.

4. **Positive Transfer:** It is associated with statements that involve two containers. It states a transfer of the quantity of entities from second container to the first. Hence, entities are added to the first container and removed from the second, indicating both operations: addition and subtraction. Examples:

   - रजत ने उससे 41 नींबू लिए।

     rajat ne usase 41 neemboo lie.

     Gloss: Rajat took 41 lemons from him.

   - ट्रेन में 25 यात्री रेलवे स्टेशन से चढ़े।

     tren mein 25 yaatree relave steshan se chadhe.

     Gloss: 25 passengers boarded the train from the railway station.

5. **Negative Transfer:** It is associated with statements that involve two containers. It states a transfer of the quantity of entities from first container to the second. Hence, entities are added to the second container and removed from the first, indicating both operations: addition and subtraction. Examples:

   - रजत ने उसको 41 नींबू दिए।

     rajat ne usako 41 neemboo die.

     Gloss: Rajat gave 41 lemons to him.

   - ट्रेन से 25 यात्री रेलवे स्टेशन पर उतरे।

     tren se 25 yaatree relave steshan par utare.

     Gloss: 25 passengers from the train got off at the railway station.

5.2.1.1 Annotation of verbs.

In our dataset of 2336 word problems, around 1713 word problems are based on addition and subtraction operations and are eligible to be considered as a dataset for word problem solving using verb categorisation. In these 1713 word problems, there are around 200 verbs. These verbs were annotated with the categories mentioned above.

In word problems, a verb occurs in a small set of contexts, because of which we can choose to focus on the most prevalent contexts. Therefore, to keep the process simple, each instance of the verb was annotated considering its context (i.e. after analysing the entire word problem). Most verbs belonged to one category. There were a few instances in the dataset where the verb belonged to multiple categories. One such example is where the verb 'kha' (to eat) takes these different categories in different contexts in the dataset:

- In this example, 'kha' acts like a **negative** verb.

  बंदर के पास 407 मूंगफलियाँ थीं। उसने उनमें से 129 खाईं। बंदर के पास कितनी मूंगफलियाँ हैं?

  bandar ke paas 407 moongaphaliyaan theen. usane unamen se 129 khai. bandar ke paas kitanee moongaphaliyaan hain?

  Gloss: The monkey had 407 peanuts. He ate 129 of them. How many peanuts does the monkey have?

- In some other examples in the dataset like the one given below, 'kha' acts like a **positive** verb.

  बॉबी ने 26 टॉफ़ियाँ खाईं। फिर उसने 17 और खा लीं। बॉबी ने कितनी टॉफ़ियाँ खाईं?

  bobee ne 26 taaaifiyaan khaeen. phir usane 17 aur kha leen. bobee ne kitanee taaaifiyaan khaeen?

  Bobby ate 26 candies. Then he ate 17 more. How many candies did Bobby eat?

Figure 5.5 presents a graph that helps visualise the distribution of verbs in the filtered HAWP dataset in different categories:

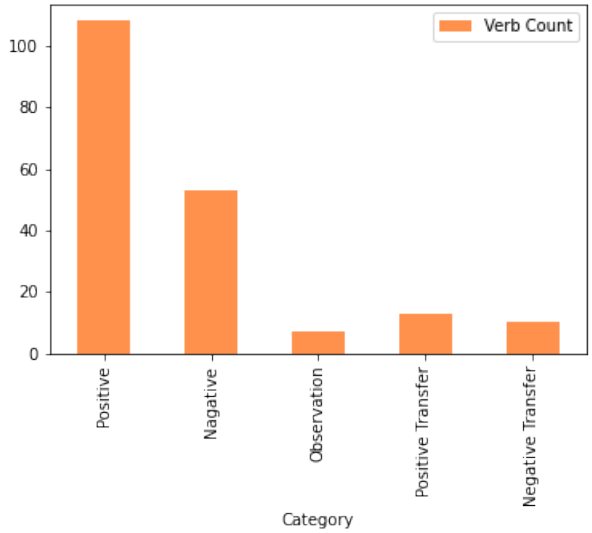


Figure 5.5: Distribution of HAWP verbs in different categories

5.2.1.2 Problems faced during annotation of verbs

Some patterns were observed in word problems (like the presence of certain words in the word problems can affect the corresponding equation or the verb category). Therefore, special attention had to be paid to these aspects and patterns, which are discussed in detail with examples below:

- Certain pairs of verbs' categories must be complement to each other, irrespective of their contexts. Examples of such verb pairs include: 'sona'-'jaagna' (sleep-wake up), 'jeetna'-'haarna' (win-lose). Since their meanings are complementary to each other, if they occur in the same word problem, they should hold complementary verb categories. For example, the table 5.2 shows the two instances of the verb 'so' (to sleep) and the one instance of 'jaag' (to wake up) in the dataset.

  Here we can observe that 'so' (to sleep) takes 2 Positive and 1 Negative categories. Hence, following the most votes rule, 'so' takes Positive and 'jaag' (to wake up) also takes Positive. However, if we do so, in the first example, the operational semantics of the verbs is not satisfied. Semantically, the two verbs are opposites and denote actions that are likely to affect the equation in opposite ways. Therefore, we must keep such kind of pairs of verbs in mind before finalising their category. For such cases, we looked at all examples of the verb in the dataset where complimentary verbs occurred together in the same problem and labeled them based on their role in those contexts.

- When words like 'wapas', 'nahi' occur before verbs, they act as the NOT operator, thus changing the original category of verbs:

Verb	Examples	Category
सो	1 छोटे शहर में 98 बिल्लियाँ हैं। अगर उनमें से 92 सो रही हैं, तो कितनी बिल्लियाँ जाग रही हैं? 1 chhote shahar mein 98 billiyaan hain. agar unamen se 92 so rahee hain, to kitanee billiyaan jaag rahee hain? Gloss: There are 98 cats in 1 small town. If 92 of them are asleep, how many cats are awake?	Negative
सो	हैरी हाउंड को कल 1 भयानक कान का दर्द हुआ। कल जब मैंने उसके कान में झाँका, तो मैंने देखा कि उसके दाहिने कान में 36 पिस्सू थे और उसके बाएँ कान में 85 पिस्सू शांति से सो रहे थे। मैंने हैरी हाउंड के कान साफ़ किए। कितने पिस्सू मारे गए? hairee haund ko kal 1 bhayaanak kaan ka dard hua. kal jab mainne usake kaan mein jhaanka, to mainne dekha ki usake daahine kaan mein 36 pissoo the aur usake baen kaan mein 85 pissoo shaanti se so rahe the. mainne hairee haund ke kaan saaf kie. kitane pissoo maare gae? Gloss: Harry the Hound got 1 horrible earache yesterday. Yesterday when I peeped into his ear, I saw that there were 36 fleas in his right ear and 85 fleas were sleeping peacefully in his left ear. I cleaned Harry the Hound's ears. How many fleas were killed?	Positive
सो	भावना आज 8 घंटे सोई और कल वह थकान होने के कारण 10 घंटे सोई थी। भावना ने इन दिनों में कुल कितने घंटों की नींद ली? bhaavana aaj 8 ghante soee aur kal vah thakaan hone ke kaaran 10 ghante soee thee. bhaavana ne in dinon mein kul kitane ghanton kee neend lee? Gloss: Bhavna slept for 8 hours today and yesterday she slept for 10 hours due to tiredness. How many hours of sleep did Bhavna take during these days?	Positive
जाग	1 छोटे शहर में 98 बिल्लियाँ हैं। अगर उनमें से 92 सो रही हैं, तो कितनी बिल्लियाँ जाग रही हैं? 1 chhote shahar mein 98 billiyaan hain. agar unamen se 92 so rahee hain, to kitanee billiyaan jaag rahee hain? Gloss: There are 98 cats in 1 small town. If 92 of them are asleep, how many cats are awake?	Positive

Table 5.2: Examples of verbs: Polar verb category

– लिनो ने सुबह समुद्र के किनारे 324 सीपियाँ उठाईं और दोपहर में 292 सीपियाँ वापस रखीं। उसके पास कितनी सीपियाँ हैं?

  lino ne subah samudr ke kinaare 324 seepiyaan uthaeen aur dopahar mein 292 seepiyaan vaapas rakheen. usake paas kitanee seepiyaan hain?

  Gloss: Lino picked up 324 shells on the beach in the morning and put 292 back in the afternoon. How many shells does he have?

– 1 कार कंपनी ने उत्तरी अमरीका में 3884 कारें बनाई, लेकिन यूरोप में 2871 कारों को वापस बुलाया गया। कितनी कारें सफलतापूर्वक बनाई गईं?

  1 kaar kampanee ne uttaree amareeka mein 3884 kaaren banaee, lekin yoorop mein 2871 kaaron ko vaapas bulaaya gaya. kitanee kaaren saphalataapoorvak banaee gaeen?

Gloss: 1 car company manufactured 3884 cars in North America, but 2871 cars were called back in Europe. How many cars were manufactured successfully?

– दिसंबर से पहले ग्राहकों ने मॉल से 1346 गुलबंद खरीदे। दिसंबर के दौरान, उन्होंने 6444 खरीदे। उसके बाद 1 भी गुलबंद नहीं बिका। ग्राहकों ने कुल कितने गुलबंद खरीदे?

disambar se pahale graahakon ne mol se 1346 gulaband khareede. disambar ke dauraan, unhonne 6444 khareede. usake baad 1 bhee gulaband nahin bika. graahakon ne kul kitane gulaband khareede?

Gloss: Before December, customers bought 1346 mufflers from the mall. During December, they bought 6444. After that not even 1 muffler was sold. How many mufflers did the customers buy in all?

- Confusion between the verb categories - Positive transfer and Negative transfer may arise for Hindi. Since Hindi has relatively free word order, the transfer of entities from first container to second and transfer from second to first would change since the word order is not fixed and hence would not be specific to the verb. Let us understand with an example: Consider चढ़े to be Positive Transfer and उतरे to be Negative Transfer.

1. रेलवे स्टेशन से 25 यात्री ट्रेन में चढ़े।

   relave steshan se 25 yaatree tren mein chadhe.

   Gloss: 25 passengers boarded the train from the railway station.

2. ट्रेन में 25 यात्री रेलवे स्टेशन से चढ़े।

   tren mein 25 yaatree relave steshan se chadhe.

   Gloss: 25 passengers boarded the train from the railway station.

Though both examples are same and the verb is also same in both examples, due to relatively free word order, in the first example entities are getting removed from the first container however in the second example entities are getting removed from the second container. Same can be observed for the verb उतरे .

1. रेलवे स्टेशन पर 25 यात्री ट्रेन से उतरे।

   relave steshan par 25 yaatree tren se utare.

   Gloss: 25 passengers got off the train at the railway station.

2. ट्रेन से 25 यात्री रेलवे स्टेशन पर उतरे।

   tren se 25 yaatree relave steshan par utare.

   Gloss: 25 passengers from the train got down at the railway station.

Since the number of such examples were relatively less in number in our dataset, we did not change the verb categories.

Once the dataset was annotated, we looked at different ways of categorising verbs. The process of building models and logic for verb categorisation is explained next.

5.2.2 Train-Test Split

The dataset was split into train and test sets using Stratified k-fold approach with k as 5. These sets were used for training and testing all the models described in this chapter.

5.2.3 Models

To predict categories of verbs, we used three methods:

1. **Closest Verb Model:** In the first method, we used the category of other verbs which are closer to the test verb to determine its category.
2. **Context Model:** In the second method, we predicted the category of verb using the grammatical information of its context.
3. **MuRIL Contextual Embeddings:** MuRIL model was fine-tuned on the verb categorization data.

5.2.4 Closest Verb Model

The basic idea behind this method is that similar or closer verbs may have the same categories. Therefore, in this method, we find the verb in the training set most proximate to the given test verb and then assign its category to the test verb. But this is not as easy as it sounds because this "similarity" or "closeness" can be defined in various ways. And while two verbs may be close in one way, they may not be as close via some other definition of "closeness". For the same reason, we tried exploring two meanings of "closeness": proximity of verbs in knowledge graphs and proximity of verbs in the vector space.

Using these two definitions, we considered the following two methods:

- Verb's Nearest Neighbour: Finding closest verb using word vectors.
- Verb's Semantic Relatives: Finding closest verb using WordNet.

5.2.4.1 Verb's Nearest Neighbour

This method of categorising verbs is motivated by the idea that words that are related to each other in meaning will have their respective word vectors closer to each other as well.

We had our doubts regarding this approach because in case of verbs opposites like 'bechna'-'khareedna' (sell-buy) or 'chadhana'-'utarna' (get on-get down) or 'khana'-'pakana' (eat-cook): the verbs in each pair may belong to different verb categories but their word vectors would be close. This is addressed further in 5.2.4.3.

5.2.4.2 Experimental Setup

Our dataset consists of word problems, verb categories annotated as part of earlier sections of this chapter and the equations to the word problems (also includes other annotations). Figure 5.6 shows the current state of our dataset.

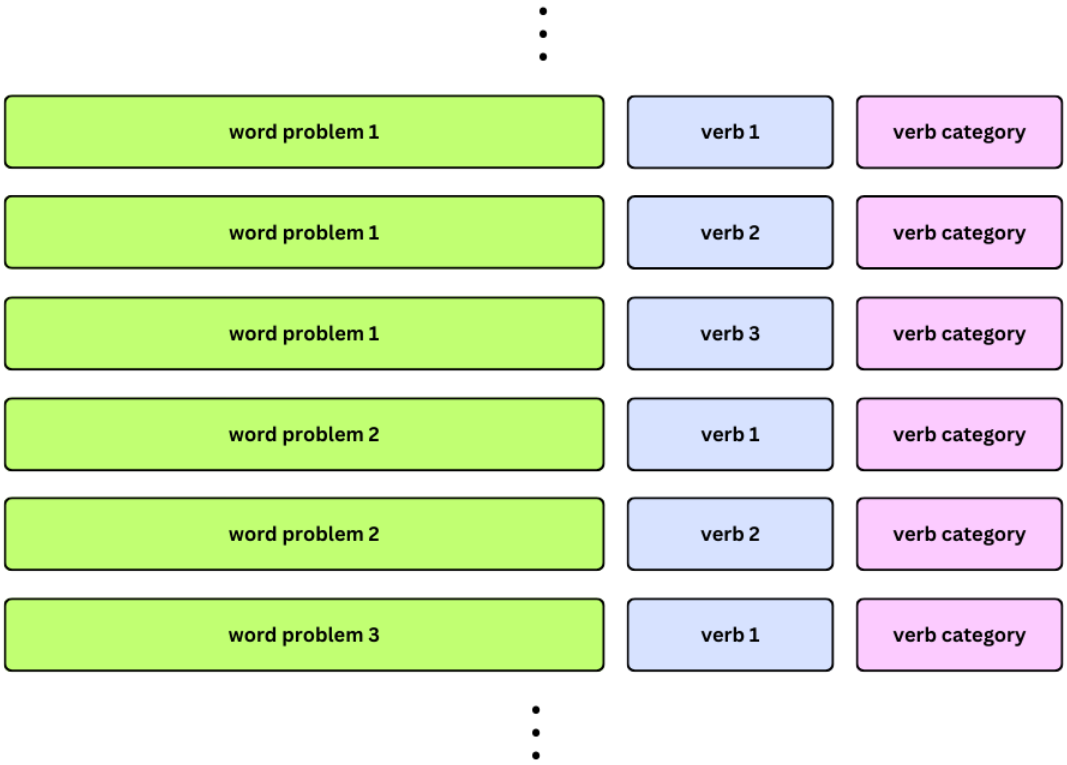


Figure 5.6: Initial state of dataset

Since we have verbs mapped to their categories in column 2 and column 3 respectively. We extracted unique verbs mapped to their categories from this data.

Next, for acquiring word vectors for these verbs, we used pre-trained word vectors [5] with dimension 300 for Hindi.

For each of the 5 folds of train-test data, the verb category corresponding to the training verb's vector which was closest to the test verb, was assigned to the test verb. Now let us look at the results of this experiment.

5.2.4.3 Results and Error Analysis

The average f1-score using this approach was found to be 0.8953. On analysing the f1-scores for each categories, we found that verbs with the actual category of Positive are classified incorrectly the most. In most folds, the verb with Positive category gets assigned to the Negative category. Whereas Negative Transfer, Positive Transfer and Observation categories are classified correctly with around 0.96 mean f1-score.

Previously, we explained our doubts about opposite verbs being classified into the same categories because of their vectors being close to each other. To address this concern, we list our findings in Table 5.3, which shows some examples of nearest verb pairs identified during the experiment, whether their verb categories matched and some observations around each pair. Though the number of unsuccessful matches is far less than successful matches, as can be confirmed from the f1-scores of this classification task, this has the potential to affect the accuracy of verb categorisation.

Verb Pair		Category Match?	Comments
Test Verb	Matched Verb	✓/ ✗	
samajh (under-stand)	soch (think)	✓	These verbs have closer meanings.
bichh (lay)	daal (put)	✓	Not exactly synonyms but these verbs have closer meanings.
jhadh (fall off)	gir (fall)	✓	Synonyms in some contexts.
khila (feed)	pila (give water to)	✓	Not exactly synonyms but these verbs have closer meanings.
sad (rot)	mar (die)	✓	Not exactly synonyms but these verbs have closer meanings.
pighal (melt)	toot (break)	✓	Not exactly synonyms but both verbs relate to destruction.
so (sleep)	jaag (wake up)	✗	Antonyms
bita (spend [time])	kama (earn)	✗	Not exactly antonyms but closer to opposite meanings
jhool (swing/ride)	gir (fall)	✗	Antonyms
bun (knit)	cheel (peel)	✗	Not exactly antonyms but closer to opposite meanings. Test verb is closer in meaning to creation while matched verb is closer to removal.

Table 5.3: Examples of verb categories found using Verb's Nearest Neighbours

5.2.4.4 Verb's Semantic Relatives

The idea behind this method was to have a rule based system that looks at the synonyms, antonyms and other "semantic relatives" of the test verb that might be present in the train set and then, make use

of the category of the train verb along with its semantic relation to the test verb to decide the category for the test verb. Therefore, a very rough idea of the algorithm would be:

foreach *testVerb* **do**
  foreach *trainVerb* **do**
    if *SemanticRelation (trainVerb, testVerb) is not null* **then**
      $current\_test\_category \leftarrow$`Rule (category [trainVerb ], SemanticRelation (trainVerb, testVerb))`
    end
  end
  $test\_category \leftarrow current\_test\_category$ with max votes /*may also add a smoking gun logic for deciding category */
  return $test\_category$
end

We tried using this method to determine verb categories but found many inconsistencies and missing relationships for the words in our dataset. For most verbs, synonyms and antonyms were missing. Therefore, due to a lack of data to successfully conduct this experiment, the technique of using Verb's Semantic Relatives was discarded.

Moreover, we identified a limitation in both methods of the Closest Verb Model: Verb's Nearest Neighbours and Verb's Semantic Relatives. Looking at the predicted verb categories and samples, we understand that the correct sense of verb gets missed in these methods because we only use verbs without their context. This way not only does the proper sense gets missed, but the verb may also get confused with a different POS category altogether. For example, the verb 'jal' (jalna) matched with the verb 'bah' (bahna). In this case, jal was taken with the sense 'water' and not 'burn'. Similarly, 'fail' (failna), which had the sense 'spill' in the word problem was matched with 'badh' (grow). This shows the sense of 'spread' associated with 'fail' must have been considered here. Since Hindi has many other words with the homonymy relation (anekaarthi shabd), this method might result in unintended errors due to a lack of context.

Due to the issues identified in these methods of verb categorisation, especially the effect of lack of context with the verb, we introduce a model that categorises verbs based on their context.

5.2.5 Context Model

This model uses the context of a verb to determine its category hence it is named context model. The idea is to use a bag of a verb and its neighbours in their actual order as a sample and the category of the verb as the label.

The tokens in a sentence can be different from a similar sentence but their role and association with the verb may be same. For example:

1. उसने कचरे के 1576 बैग उठाए।

   usane kachare ke 1576 baig uthae.

He picked up 1576 bags of garbage.

2. राजवीर ने काग़ज़ के 3 टुकड़े उठाए।

   raajaveer ne kaagaj ke 3 tukade uthae.

   Rajveer picked up 3 pieces of paper.

3. उसने पक्षियों की 532 प्रजातियाँ देखीं।

   usane pakshiyon kee 532 prajaatiyaan dekheen.

   He saw 532 species of birds.

Therefore, instead of taking the tokens of neighbours as is, we used their Part of Speech (POS) and Universal Dependency (UD) tags to create a level of abstraction in the context while still preserving the grammatical: morphological and syntactic information. So now, even if tokes are different their POS/UD would help finding similarities in the verb contexts and making use of them to identify the verb category.

Hence, our desired samples look something like Figure 5.7.

5.2.5.1 Data Processing

Looking back at 5.6, one can easily spot that there is a gap in our current state and desired state of data shown in Figure5.7. Therefore, we had to process this data for it to be suitable for the MWP verb classification task.

We used in-house tool to get the root and Part of Speech (POS) of each token in each sentence of each word problem in the dataset. We used ISC-parser from Natural language tool-kit for Indian Language Processing[1] to get Universal Dependency tags of each token in each sentence of each word problem in the dataset.

Using this morph-level information i.e Part of Speech (POS) tags and syntax-level i.e. Universal Dependency (UD) tags, we created samples for the task. Moreover, after a number of experiments we finalised the size of the context window as 7 i.e. the number of neighbours to the left and right of the verb are 3 - therefore we will have 3 neighbours to the right of the verb and 3 neighbours to the left of the verb i.e.

$$v-3, v-2, v-1, v, v+1, v+2, v+3$$

where v represents the index of the verb in a sentence. Since Hindi follows a SOV (Subject-Object-Verb) order, there many be many instances where the neighbours to right of verb do not exist as the sentence ends at the verb. For those cases, we added a constant in place of the missing neighbours.

[1] `https://github.com/iscnlp/iscnlp/tree/master/iscnlp`

Given the SOV nature of Hindi, it might seem like a good idea to only consider the left of the verb in the samples. This experiment was also conducted as part of those performed to find the best number of neighbours, but the idea was dropped as it did not produce the best results.

Finally, for each verb token of each sentence of each word problem, we created the sample using the POS and UD of two neighbours on both sides for our final data to look something like Figure 5.8.

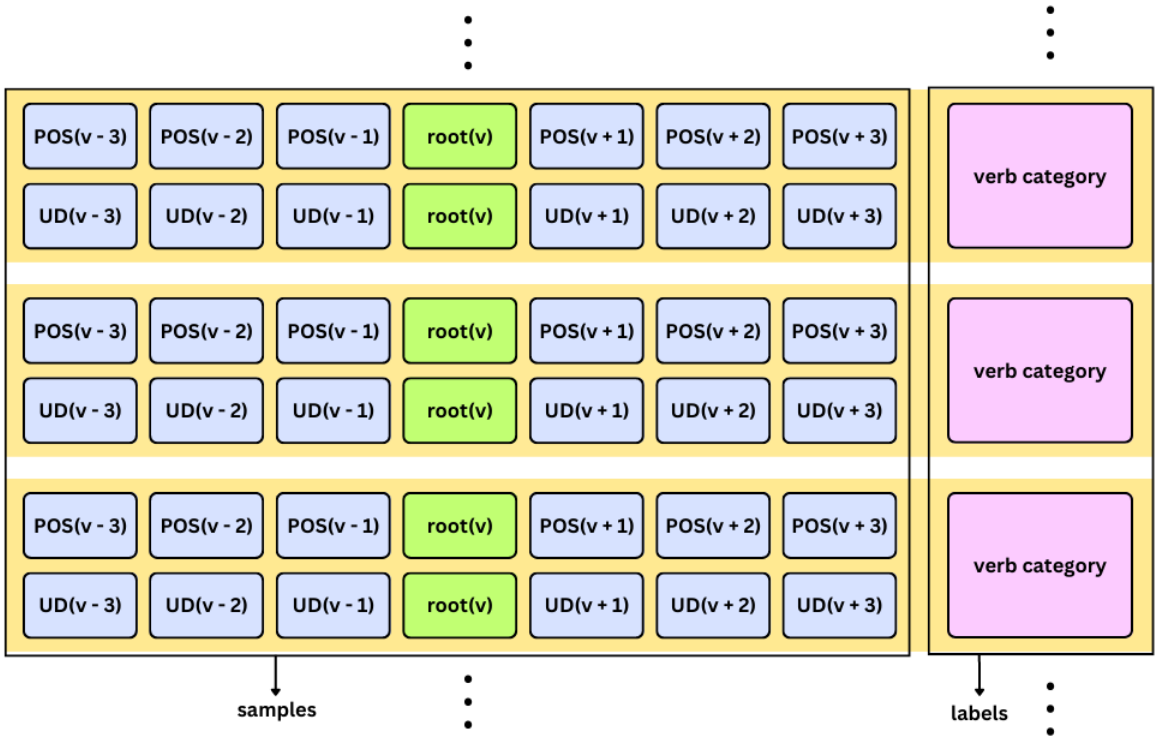


Figure 5.8: Final state of verb categorisation dataset

This process was followed for train and test sets of all five folds.

Note: While annotating verbs, along with the five verb categories mentioned before, we used one more category - 'NA'. This category was used to eliminate the words that were incorrectly labelled as verbs by POS taggers or dependency parsers. Doing this, would clear incorrect examples of verbs that belonged to a particular category and would only present the model with correct data.

5.2.5.2 Experimental Setup

We adopted a 5-Fold cross validation technique for evaluating all the verb categorization models including the Context Model. We performed this classification task using 3 Machine Learning approaches:

- Logistic Regression

- Random Forest
- Support Vector Machines (SVM)

Before feeding the data to the models, the train and test samples were vectorised using TF-IDF which stands for Term Frequency – Inverse Document Frequency and is commonly used to represent textual documents as vectors.

5.2.5.3 Results and Error Analysis

Table 5.4 shows the results of Context Model for verb categorisation task.

Approach	**F1-score**
Linear Regression	0.8645
Random Forest	0.8828
Support Vector Machines	0.9036

Table 5.4: Average scores using various ML approaches

SVM classified Positive Transfer, Negative Transfer and Observation verbs with approximately 0.99 accuracy. While the same is observed for Random Forest and Logistic Regression for the Observation category, the two models gave relatively less accurate predictions for Positive and Negative transfers.

On analysing the errors made by SVM, we identified that verbs with the actual category of Positive and Negative are classified incorrectly the most in that order. It appears that the model gets confused between the two categories for some examples. Similar confusion but to a higher degree was seen in the results given by the Logistic Regression model. For Random Forest, the error in predicting Negative as Positive was comparatively much higher than its confusion in any other category predictions, followed by Positive being predicted as Negative. Overall, all models got confused between the Positive and Negative categories.

Moreover, the most incorrect predictions that are made are of the categories: Positive and Negative. This can be attributed to more number of samples for the two in the dataset, as seen in Figure 5.5.

To understand the errors, we looked at the samples that failed. The contexts of the verbs in the failed samples were so common that these could be mapped to verbs with different categories. The underlined context in the following word problems is similar, although they cover different verbs with different categories:

1. बेका अपनी चाची से मिलने के लिए 873 मील उड़ी। जैक्सन अपनी चाची से मिलने के लिए 563 मील उड़ा। जैक्सन की तुलना में बेका कितने ज़्यादा मील उड़ी?

   beka apanee chaachee se milane ke lie 873 meel udee. jaiksan apanee chaachee se milane ke lie 563 meel uda. jaiksan kee tulana mein beka kitane zyaada meel udee?

Becca flew 873 miles to visit her aunt. Jackson flew 563 miles to visit his aunt. How many more miles did Becca fly than Jackson?

2. दिसंबर से पहले ग्राहकों ने मॉल से 6444 गुलबंद खरीदे। दिसंबर के दौरान उन्होंने पहले की तुलना में 1346 कम गुलबंद खरीदे। दिसंबर में कितने गुलबंद बेचे गए?

   disambar se pahale graahakon ne mol se 6444 gulaband khareede. disambar ke dauraan unhonne pahale kee tulana mein 1346 kam gulaband khareede. disambar mein kitane gulaband beche gae?

   Before December, customers bought 6444 mufflers from the mall. During December, they bought 1346 less mufflers than before. How many mufflers were sold in December?

3. दीवाली पर पारुल ने 57 दीपक जलाए। उनमें से 15 दीपक बुझ गए। बताइए, अब कितने दीपक जलते हुए रहे?

   deevaalee par paarul ne 57 deepak jalae. unamen se 15 deepak bujh gae. bataie, ab kitane deepak jalate hue rahe?

   Parul lit 57 lamps on Diwali. Out of them, 15 lamps went off. How many lamps were still lit?

4. संग्रहालय का दौरा करने के बाद, बंटी अपने होटल में वापस आया। वहाँ पहुँचने के लिए, वह रॉकफेलर सेंटर तक 354 कदम चला और फिर टाइम्स स्क्वायर तक 228 कदम चला। होटेल पहुँचने से पहले वह कितने कदम चला?

   sangrahaalay ka daura karane ke baad, bantee apane hotal mein vaapas aaya. vahaan pahunchane ke lie, vah rokaphelar sentar tak 354 kadam chala aur phir taims skvaayar tak 228 kadam chala. hotel pahunchane se pahale vah kitane kadam chala?

   After visiting the museum, Bunty came back to his hotel. To get there, he walked 354 steps to Rockefeller Center and then 228 steps to Times Square. How many steps did he walk before reaching the hotel?

Table 5.5 shows the contexts of these examples i.e. corresponding samples fed to the model. 'Failed sample' denotes a test sample and 'Similar contexts' denotes training samples having similar contexts to that of test sample.

	Sample	Predicted Category	Actual Category
Failed Sample	PSP, QT_QTC, N_NN, उड़ा , RD_PUNC, c, c, case, nummod, dobj, उड़ा , punct, c, c	N	P
Similar Contexts	PSP, QT_QTC, N_NN, खरीद , RD_PUNC, c, c, case, nummod, dobj, खरीद , punct, c, c	-	T+
	PSP, QT_QTC, N_NN, जला , RD_PUNC, c, c, case, nummod, dobj, जला , punct, c, c	-	N
	PSP, QT_QTC, N_NN, चल , RD_PUNC, c, c, case, nummod, dobj, चल , punct, c, c	-	P

Table 5.5: Examples of contexts common over multiple categories [Please find gloss below]

Gloss for Table 5.5:

- c: constant
- N: Negative
- P: Positive
- T+: Positive Transfer

5.2.6 MuRIL Contextual Embeddings

Contextual embeddings, especially BERT [4] based embeddings, have been shown to be very effective for classification as well as generalization tasks. BERT is trained in two stages: pre-training and fine-tuning. The model is first trained on a huge monolingual corpus to learn language-specific representations and then fine-tuned on a downstream task. In our case, the downstream task is the verb categorization task. As this is a classification task, it is a perfect test for using BERT or BERT-like models. For this, we used MuRIL [12], a multilingual transformer [30] model trained on English and 16 Indian languages. MuRIL is pre-trained using masked language modelling as well as translation language modelling. It has a combined vocabulary of 197K words.

5.2.6.1 Data Preparation

We used the same dataset as used for the machine learning (ML) models. For the ML models, each sample consists of a verb token and its neighboring words. The size of the context window is 7. POS and universal dependency features in a context window around the verb are also embedded. As MuRIL can

handle large contexts, we do not limit ourselves to a fixed context window. For this task, all the words till a verb is encountered constitute a sample. A total 6506 samples were created for verb categorization. Let us take an example to understand this better.

- Original Question:
  कनिष्क को समुद्र तट पर 47 सीपियाँ मिलीं, उसने लैला को 25 सीपियाँ दीं। उसके पास अब कितनी सीपियाँ हैं?

  Gloss: Kanishk found 47 shells on the beach, he gave 25 shells to Laila. How many shells does he have now?

- Samples for Verb Categorization

  – कनिष्क को समुद्र तट पर 47 सीपियाँ मिलीं

    Gloss: Kanishk (found) 47 shells on the beach

  – कनिष्क को समुद्र तट पर 47 सीपियाँ मिलीं , उसने लैला को 25 सीपियाँ दीं ।

    Gloss: Kanishk found 47 shells on the beach, he (gave) 25 shells to Laila.

  – उसके पास अब कितनी सीपियाँ हैं ?

    Gloss: How many shells does he (have) now?

5.2.6.2 Experimental Setup

MuRIL has 236 million parameters and it uses AdamW [19] optimizer. We used 5-fold cross validation technique to evaluate the model. MuRIL was fine tuned for 10 epochs with a batch size of 4.

Approach	F1-score
MuRIL Fine-tuning	**0.962**

Table 5.6: Verb Categorization Results with MuRIL Fine Tuning

5.2.6.3 Results and Discussion

The results are shown in table 5.6. We can observe that MuRIL fine-tuning outperforms all the ML model significantly. We record an improvement of 6% from the SVM model. Maximum number of errors are attributed to the "NA" category. As the number of samples for this category is very small, the model often gets confused with the other classes. Apart from this category, we did not observe any significant errors in other categories.

5.3 WPS Solver using Verb Categories

Now that we have categorised our verbs to help us identify the operations to be used to solve the word problems, we move on to explain the second and final step i.e. building a solver which will take the word problems and return the answer to word problems.

5.3.1 Setup

We built a simple rule based system that takes in word problems and generates answers to these word problems.

This section focuses on how we built the solver by visualising word problems as states and how transitions between states were carried out using verb categories. The basic structure and components of a state will be explained next.

5.3.1.1 States and their components

While building the solver, we revisited and redefined what a state is and its structure based on the ease with which its structure will help us solve the word problems. A state in our system consists of a container, a quantity and an entity which may or may not be accompanied by an attribute.

Things to Note:

- The states used in this section are not the same as the states used to explain the basic idea behind verb categorisation in 5.1. States as part of this system are not temporal or related to each other. They are only used to store the status of an entity i.e. its quantity and its container. Figure 5.9 shows what a state looks like for the rest of this chapter.

- In the course of this rule based system, we look at categories/operations in two different ways:

  – Verb Categories: These determine how the quantities change in states depending on the verb, depicting the number of quantities in a container.

  – Operation in Question: These determine how the quantities in state need to be operated on to get the desired answer. These are dependent on the question and can be deduced from keywords used in questions. We have not trained a model for predicting which keywords indicate which operations. Instead, we studied the patterns in questions of the word problems to identify a list of words that indicate operation in question. Let us call these as Negative Indicators and Positive Indicators. As the name suggests, Negative Indicators are words that indicate subtraction of quantities. Similarly, Positive Indicators indicate addition of quantities to generate answer to question. Examples include:

    * Positive Indicators: 'kul', 'milakar', 'milkar' etc.
    * Negative Indicators: 'mukable', 'tulna', 'pehle', 'chahiye' etc.

In word problems it is common for the order to be container, quantity, entity, verb. Hence, a state gets created each time an entity is found in a sentence of a word problem.

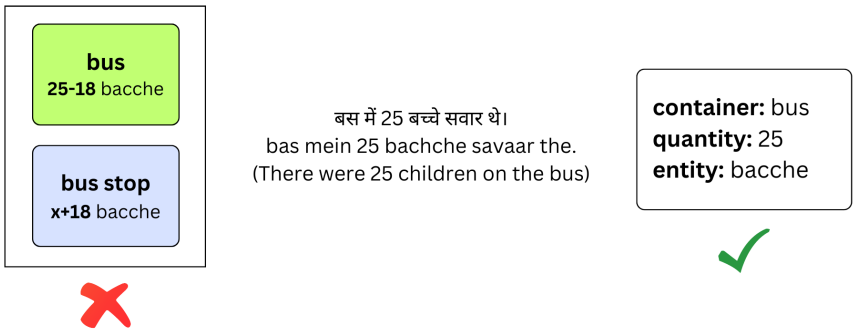


Figure 5.9

We start by iterating through all word problems. For each problem, we iterate through all its sentences and the following rules are used to extract the components of a state from the sentences:

1. A container is a proper noun or adverb of place. For example, शॉन (Shawn) in this sentence शॉन के पास 13 ब्लॉक थे। (shon ke paas 13 blok the. , Gloss: Shawn had 13 blocks.) and कटोरे (bowl) in 1 कटोरे में 95 समोसे हैं। (1 katore mein 95 samose hain., Gloss: There are 95 samosas in 1 bowl.)

2. A quantity is a number. Whenever a quantity is found, the last identified container is associated with this quantity.

3. An entity is a noun. If there is an adjective associated with this entity, it is clubbed with the entity. When a word problem has the Rupee symbol (₹), the entity is taken to be this symbol itself.

रम्या के पास 16 लाल पेंसिलें थीं। रम्या ने 7 पेंसिलें खो दीं। अब रम्या के पास कितनी लाल पेंसिलें हैं?
ramya ke paas 16 laal pensilen theen. ramya ne 7 pensilen kho deen. ab ramya ke paas kitanee laal pensilen hain?

Components of a state:

ramya ke paas 16 laal pensilen theen.
NNP QC JJ NN

Container Quantity Attribute Entity

Figure 5.10: Components of a state: Container, Entity, Quantity

5.3.1.2 Storing States

1. Once an entity is found, the associated quantity, and container are used to form a state.

रम्या के पास 16 लाल पेंसिलें थीं। रम्या ने 7 पेंसिलें खो दीं। अब रम्या के पास कितनी लाल पेंसिलें हैं?
ramya ke paas 16 laal pensilen theen. ramya ne 7 pensilen kho deen. ab ramya ke paas kitanee laal pensilen hain?

States from statements:

ramya ke paas 16 laal pensilen theen.
NNP QC JJ NN

container quantity attribute entity

container: ramya
quantity: 16
entity: laal pensil

Figure 5.11: Example of a state for a word problem

2. A list of states is maintained in which all states created as part of the previous step are stored.

3. **Handling Negative Category:** Before storing quantity in a state, if the verb that follows the identified entity of this state has its verb category as NEGATIVE, the quantity is negated and stored.

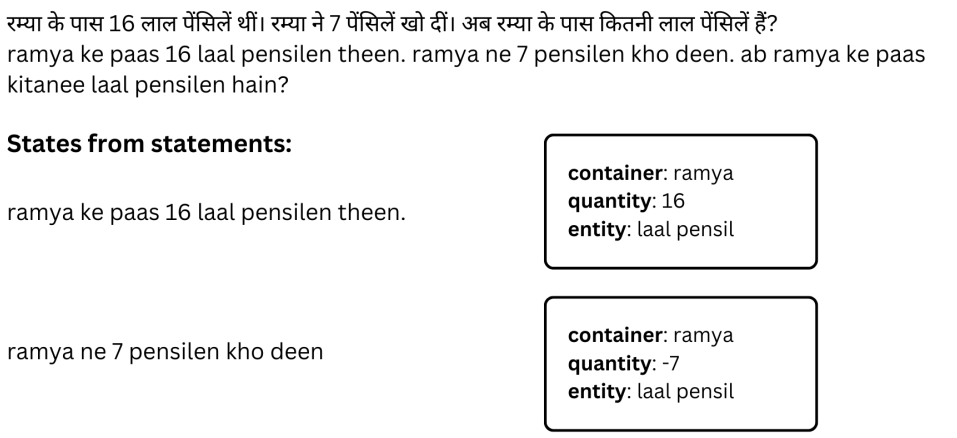


Figure 5.12: Example of Handling Negative Category

4. **Handling Transfer Category:** Once a verb is found in the word problem, we check for TRANSFER categories. We check if the verb belongs to POSITIVE TRANSFER or NEGATIVE TRANSFER category from our verb categorisation exercise.

   If a transfer category is found we updated the values of the identified states based on the transfer.

   - First, we make use of case markers to identify transfer-container1 (karta (nominative case or subject)), transfer-entity (karma (Accusative case or object)) and transfer-container2 (sampradaan (dative case or recipient)/aapadaan (ablative case)) in the sentence where transfer verb category is found. We also checked for cases where cases are included as morphemes i.e. ‘usko’, ‘use’, ‘usne’ etc.

   - We also identity the quantity of entities transferred.

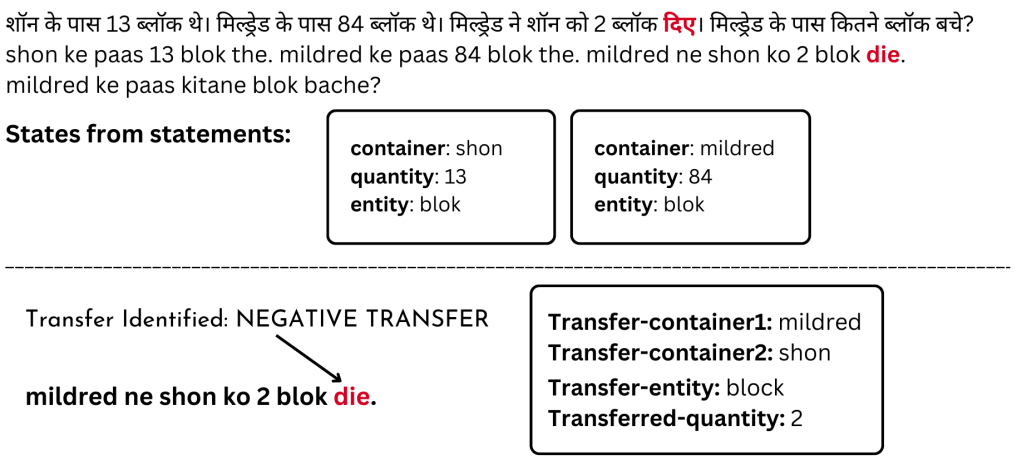


Figure 5.13: Identifying transfer components

- Then we iterate through the states and find which states have transfer-container1 and transfer-container2. Then we check if transfer-entity is present in these states.

Glossary:

- ct1_states: Represent the states which have transfer-container1 as a container.
- ct2_states: Represent the states which have transfer-container2 as a container.
- ct1_present: Represents a Boolean value indicating the presence of transfer-container1 in a state.
- ct2_present: Represents a Boolean value indicating the presence of transfer-container2 in a state.
- ct1_entity_present: Represents a Boolean value indicating the presence of transfer-entity in ct1_states.
- ct2_entity_present: Represents a Boolean value indicating the presence of transfer-entity in ct2_states.
- transferred_quantity: Represents the quantity of entities transferred between transfer-container1 and transfer-container2.

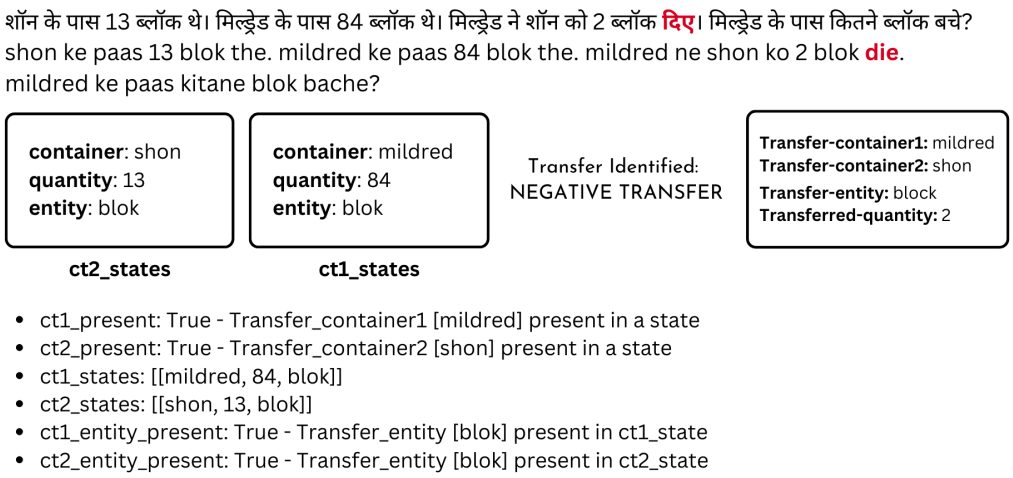


Figure 5.14: Finding the transfer components (transfer containers, transfer entity) in states for verb category: Negative Transfer

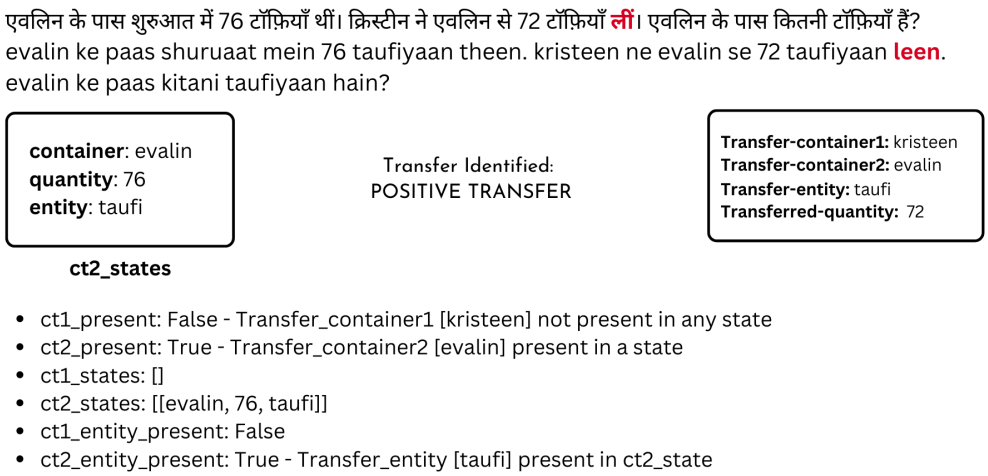


Figure 5.15: Finding the transfer components (transfer containers, transfer entity) in states for verb category: Positive Transfer

- Next, based on verb category i.e POSITIVE TRANSFER or NEGATIVE TRANSFER, we update the quantity in states using the rules given below.

  If verb category is **NEGATIVE TRANSFER**:

  (a) If ct1_present, ct2_present, ct1_entity_present and ct2_entity_present are True, we subtract the transferred_quantity from the state in ct1_states which has transfer-entity and add the transferred_quantity to the state in ct2_states which has transfer-entity.

(b) Else if ct1_present and ct1_entity_present are True, we subtract the transferred_quantity from the state in ct1_states which has transfer-entity and create a new state with transfer-container2, transferred_quantity and transfer-entity. Why? Because we did not have any states with transfer-container2 and we need to store this information that transfer-container2 has seen a gain of transferred_quantity of transfer-entity.

(c) Else if ct2_present and ct2_entity_present are True, we add the transferred_quantity to the state in ct2_states which has transfer-entity and create a new state with transfer-container1, -1 * transferred_quantity and transfer-entity. Why? Because we did not have any states with transfer-container1 and we need to store this information that transfer-container1 has seen a loss of transferred_quantity of transfer-entity.

(d) Finally, if none of the above conditions is true, which implies we do not have any states with transfer-container1 and transfer-entity in same state nor transfer-container2 and transfer-entity in same state, we create two new states, one with transfer-container2, transferred_quantity and transfer-entity and another with transfer-container1, -1 * transferred_quantity and transfer-entity.

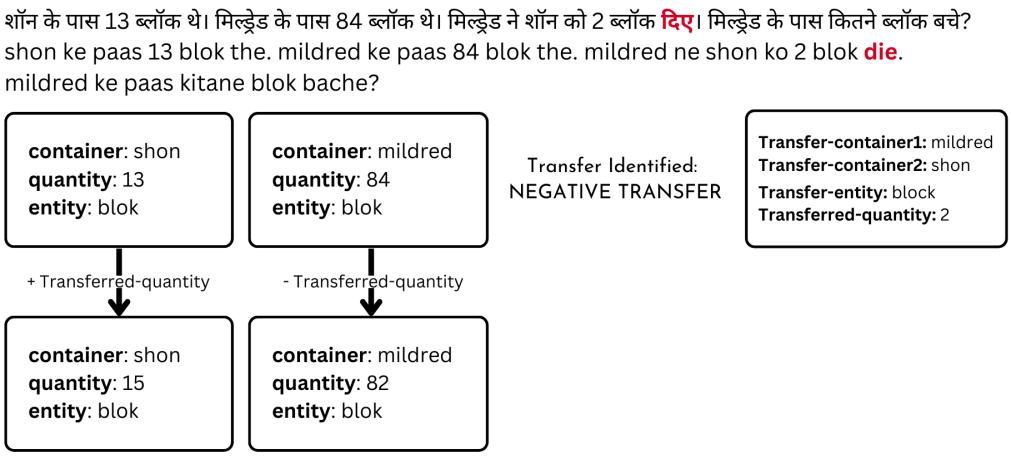


Figure 5.16: Updating states: This example follows condition (a) of Negative Transfer

Similarly, If verb category is **POSITIVE TRANSFER**:

(a) If ct1_present, ct2_present, ct1_entity_present and ct2_entity_present are True, we add the transferred_quantity to the state in ct1_states which has transfer-entity and subtract the transferred_quantity from the state in ct2_states which has transfer-entity.

(b) Else if ct1_present and ct1_entity_present are True, we add the transferred_quantity to the state in ct1_states which has transfer-entity and create a new state with transfer-container2, -1 * transferred_quantity and transfer-entity. Why? Because we did not have any states with transfer-container2 and we need to store this information that transfer-container2 has seen a loss of transferred_quantity of transfer-entity.

(c) Else if ct2_present and ct2_entity_present are True, we subtract the transferred_quantity from the state in ct2_states which has transfer-entity and create a new state with transfer-container1, transferred_quantity and transfer-entity. Why? Because we did not have any states with transfer-container1 and we need to store this information that transfer-container1 has seen a gain of transferred_quantity of transfer-entity.

(d) Finally, if none of the above conditions is true, which implies we do not have any states with transfer-container1 and transfer-entity in same state nor transfer-container2 and transfer-entity in same state, we create two new states, one with transfer-container1, transferred_quantity and transfer-entity and another with transfer-container2, -1 * transferred_quantity and transfer-entity.

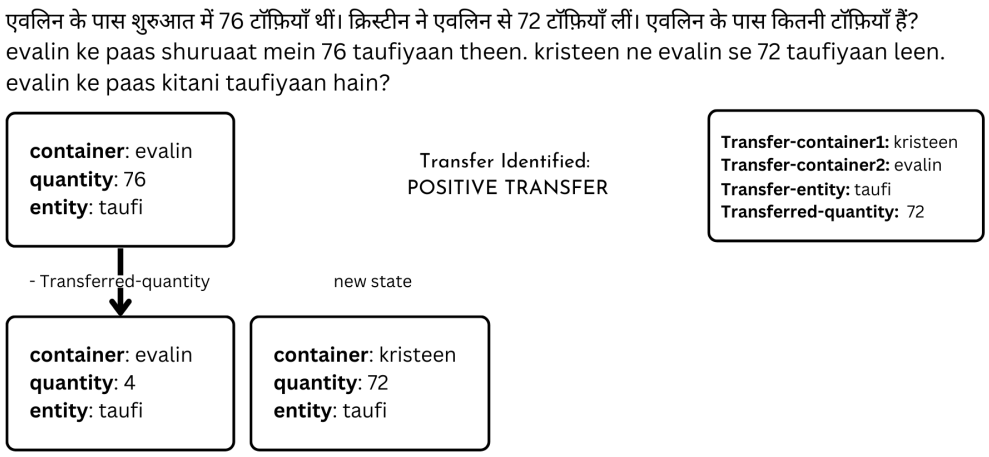


Figure 5.17: Updating states: This example follows condition (c) of Positive Transfer

After this, we mark the main operation of this word problem as a TRANSFER. This will be used while generating the answer to the word problem.

5.3.1.3 Finding Answer to Word Problems

To find the answer to a word problem, we make use of the main operation of the word problem which has been mentioned briefly towards the end of the previous section. The main operation of word problem is identified based on the following conditions checked in the mentioned order:

1. If a transfer verb category is encountered, main operation is Transfer.
2. If the operation in question is positive, main operation is Positive.
3. If the operation in question is negative, main operation is negative.
4. If none of the above conditions are met, the main operation is positive.

Here, we will look at the final sentence of the word problem which is the question.

1. First, we find the question entity and question container using the same logic as we used while creating states.

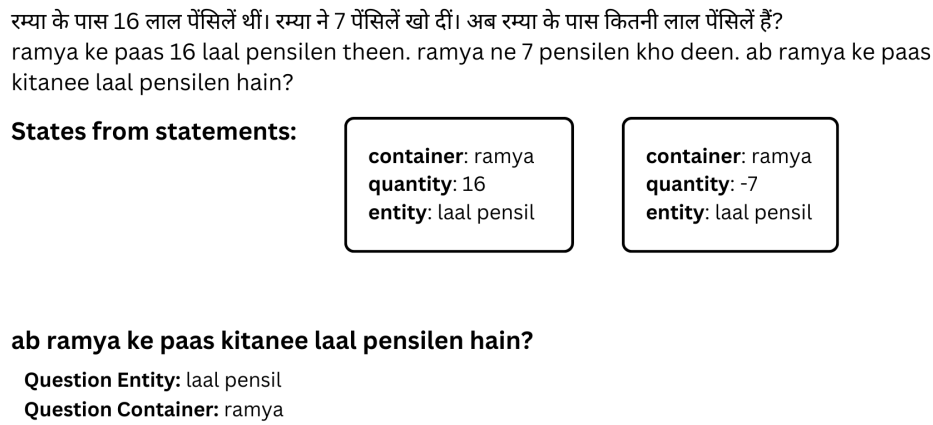


Figure 5.18: Identifying question entity and question container in MWP

2. Next, we identify the operation in question using Positive and Negative Indicators. Based on the identified operation in question, we update the main operation of word problem.

3. **Transfer:** When the main operation is TRANSFER, our calculation is already complete and we just need to find the state which has the answer to the question. Therefore, we look at all the states we created and whichever state matches the question's container, entity pair, we return its quantity as the answer.

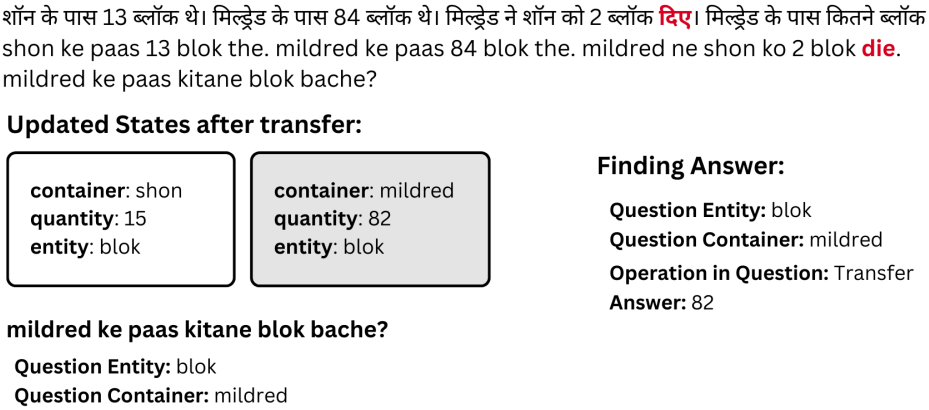


Figure 5.19: Finding Answer to Word Problem whose main operation is Transfer

4. **Negative:** When the main operation is NEGATIVE, while matching the question entity to entity in each state, we keep the quantity in the first state as is and start subtracting the quantities of the states that follow from it to finally reach our answer.

   Here is an example of a Negative Indicator ' तुलना ' ('tulna') to show how Operation in Questions are helping the solver reach the answer. Whenever a comparison occurs in a word problem, **in most cases**, it indicates a specific operation. In cases where comparison is part of question, the operation involved is NEGATIVE. Hence, it is marked as a NEGATIVE indicator. In the following examples of the this case, ' तुलना '('tulna') comes in question and indicates subtraction of relevant quantities in the word problem. This happens irrespective of whether it is accompanied by "more" (zyaada, adhik) or "less" (kam):

   (a) राल्फ के पास 50 पन्नो की 1 किताब है। उसके पास जंगली जानवरों की 26 तस्वीरें हैं। राल्फ के दोस्त डेरिक के पास जंगली जानवरों की 34 तस्वीरें हैं। डेरिक की तुलना में राल्फ के पास जंगली जानवरों की कितनी कम तस्वीरें हैं?

   raalph ke paas 50 panno kee 1 kitaab hai. usake paas jangalee jaanavaron kee 26 tasveeren hain. raalph ke dost derik ke paas jangalee jaanavaron kee 34 tasveeren hain. derik kee tulana mein raalph ke paas jangalee jaanavaron kee kitanee kam tasveeren hain?

   Ralph has 1 book of 50 pages. He has 26 pictures of wild animals. Ralph's friend Derrick has 34 pictures of wild animals. How many fewer pictures of wild animals does Ralph have than Derrick?

   States from statements:

container: raalph **quantity**: 50 **entity**: panna	**container:** raalph **quantity:** 1 **entity:** kitab	**container:** ralph **quantity:** 26 **entity:** tasveer	**container:** derik **quantity:** 34 **entity:** tasveer

   Finding Answer:

   Question Entity: tasveer
   Negative Indicator: tulna
   Operation in Question: Negative
   Answer: 34 - 26 = 8

Figure 5.20: Example of Negative Indicator: 'tulna'

   (b) वर्णिका 19.833333333333332 मील दौड़ी और 9.166666666666666 मील चली। वर्णिका पैदल चलने की तुलना में कितना ज़्यादा भागी?

   varnika 19.833333333333332 meel daudee aur 9.166666666666666 meel chalee. varnika paidal chalane kee tulana mein kitana zyaada bhaagee?

   Varnika ran 19.833333333333332 miles and walked 9.1666666666666666 miles. How much more did Varnika run than walk?

States from statements:

container: varnika
quantity: 19.8333
entity: meel

container: varnika
quantity: 9.1666
entity: meel

Finding Answer:

Question Entity: bhagi (⚠due to tagger error)
Negative Indicator: tulna
Operation in Question: Negative
Answer: 19.83 - 9.16 = 10.66

Figure 5.21: Example of Negative Indicator: 'tulna'

Another example where main operation is negative is given below. Here, the negative indicator 'pahale' is identified.

राम ने 6 गुब्बारे खो दिए। अब उसके पास 2 गुब्बारे बचे हैं। राम के पास पहले कितने गुब्बारे थे?
raam ne 6 gubbaare kho die. ab usake paas 2 gubbaare bache hain. raam ke paas pahale kitane gubbaare the?

States from statements:

container: ram
quantity: -6
entity: gubbaare

container: ram
quantity: 2
entity: gubbaare

Finding Answer:

Question Entity: gubbaare
Negative Indicator: pahale
Operation in Question: Negative
Answer: -6 - 2 = -8, take absolute: abs(-8) = 8

Figure 5.22: Example of Negative Indicator: 'pahale'

5. **Positive:** When the main operation is POSITIVE, while matching the question entity to entity in each state, we add the quantities of all the states to finally reach our answer.

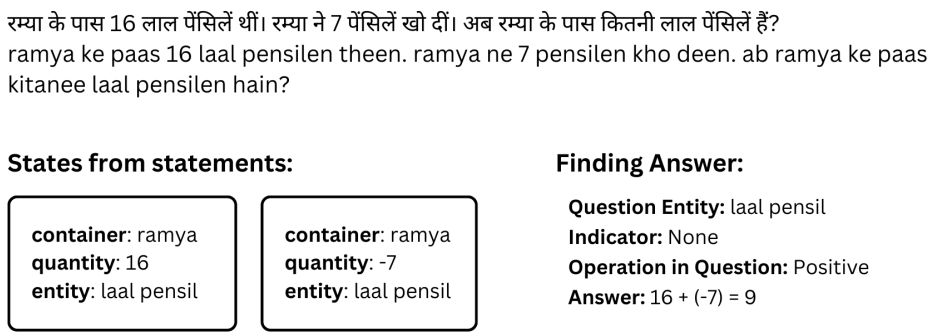


Figure 5.23: Finding answer to MWP whose main operation is Positive

5.3.1.4 Some Case-Specific Rules

We also included the following rules as part of the above explained algorithm:

- When mapping question entity to entity in states to identify relevant entities, if either question entity or states' entity is missing attribute but entity is same, the entity is counted as a relevant quantity. This can be seen through the example in Figure 5.23.

- If the entity in word problem is found to be one of 'paisa', 'keemat', 'laagat', 'rupay', we change it to ₹ so that there is consistency in the usage of generic monetary terms and if these terms are found in the question then the symbol can be mapped to the entity in the states derived from statements. Otherwise, the system will be unable to find the entity in question in the states. **Example:**

  राजबीर ने अपना मैदान की घास काटने का व्यवसाय शुरू किया। वसंत में उसने घास काटकर 2 रुपये कमाए और गर्मियों में उसने 27 रुपये कमाए। अगर उसे नई घास बिछाने के 5 रुपये और मिले, तो उसके पास कितना पैसा था?

  raajabeer ne apana maidaan kee ghaas kaatane ka vyavasaay shuroo kiya. vasant mein usane ghaas kaatakar 2 rupaye kamae aur garmiyon mein usane 27 rupaye kamae. agar use naee ghaas bichhaane ke 5 rupaye aur mile, to usake paas kitana paisa tha?

  Rajbir started his own lawn mowing business. In spring he earned Rs 2 by cutting grass and in summer he earned Rs 27. If he got Rs 5 more for laying new grass, how much money did he have?

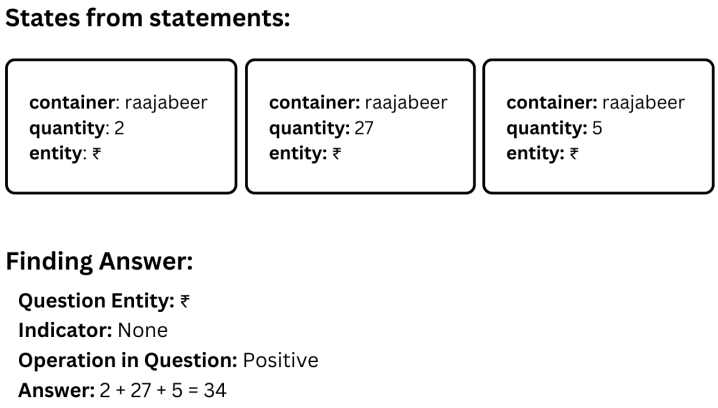


Figure 5.24: Example when the entity is related to money

- If an entity or container is not found but a quantity is found, we use the same entity and container from last state and create a new state with the quantity found. **Example:**

  गुरप्रीत के पास 26 पालतू मछलियाँ थीं। उसकी बिल्ली 6 खा गई। अब गुरप्रीत के पास कितनी मछलियाँ हैं?

  gurapreet ke paas 26 paalatoo machhaliyaan theen. usakee billee 6 kha gaee. ab gurapreet ke paas kitanee machhaliyaan hain?

  Gurpreet had 26 pet fish. Her cat ate 6. How many fish does Gurpreet have now?

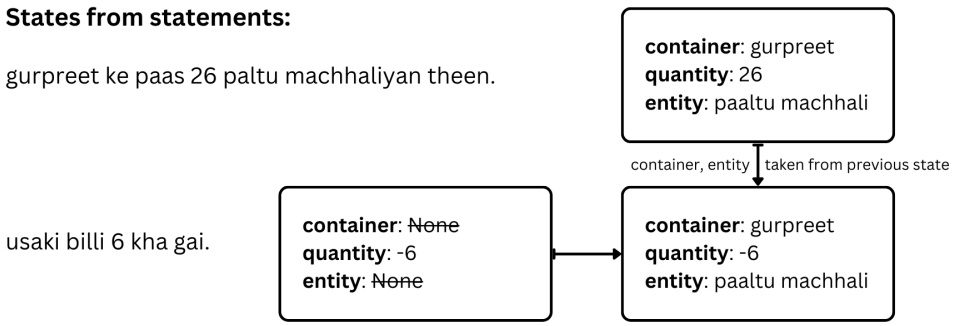


Figure 5.25: Example when the entity or container is missing in statements

- If the entity in question is not found in states, we assume the entity of the first state to be the entity in question and perform the steps of finding the answer. **Example:**

  वर्णिका 19.833333333333332 मील दौड़ी और 9.166666666666666 मील चली। वर्णिका पैदल चलने की तुलना में कितना ज़्यादा भागी?

varnika 19.833333333333332 meel daudee aur 9.166666666666666 meel chalee. varnika paidal chalane kee tulana mein kitana zyaada bhaagee?

Varnika ran 19.833333333333332 miles and walked 9.1666666666666666 miles. How much more did Varnika run than walk?

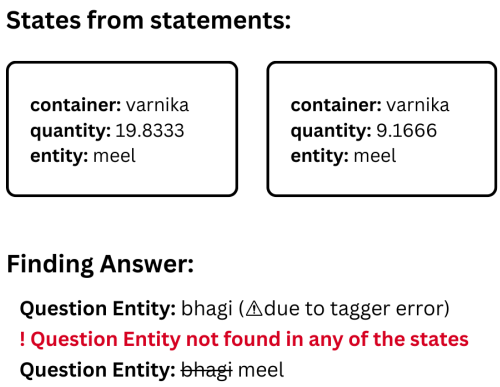


Figure 5.26: Example when an entity is not found in question

- If the final answer calculated by solver is negative, we return its absolute value. **Example:**

भावना आज 8 घंटे सोई और कल वह थकान होने के कारण 10 घंटे सोई थी। भावना ने इन दिनों में कुल कितने घंटों की नींद ली?

bhaavana aaj 8 ghante soee aur kal vah thakaan hone ke kaaran 10 ghante soee thee. bhaavana ne in dinon mein kul kitane ghanton kee neend lee?

Bhavna slept for 8 hours today and yesterday she slept for 10 hours due to tiredness. How many hours of sleep did Bhavna take during these days?

States from statements:

container: bhavana **quantity:** -8 **entity:** ghante	**container:** bhavana **quantity:** -10 **entity:** ghante

Finding Answer:

Question Entity: ghante
Positive Indicator: kul
Operation in Question: Positive
Answer: (-8) + (-10) = ~~−18~~ take absolute: abs(-18) = 18

Figure 5.27: Example when the calculated answer is found to be negative

5.3.2 Evaluation and Results

As it is clear from the explanation of the solver, instead of equations, answers are calculated by this rule based system.

We evaluated the solver in two ways:

1. The solver was tested on test sets using verb categories predicted in Section 5.2 and an average accuracy of 37.8% was reported over the 5 sets.

2. The solver was tested on the entire filtered HAWP dataset using actual verb categories and an accuracy of 37.67% was reported.

5.3.3 Error Analysis and Limitations of Solver

We analysed the errors of solver and present our findings below:

- Irrelevant Information: The solver fails to identify some cases of irrelevant information.
  - राम इस महीने 11 क्रिकेट के मैच देखने गया। वह पिछले महीने 17 मैच देखने गया था और अगले महीने वह 16 मैच देखने जाएगा. वह कितने मैच देख चुका है?

    raam is maheene 11 kriket ke maich dekhane gaya. vah pichhale maheene 17 maich dekhane gaya tha aur agale maheene 16 maich dekhane jaaega. vah ab tak kul kitane maich dekh chuka hai?

    Ram went to watch 11 cricket matches this month. He went to watch 17 matches last month and next month he will go to watch 16 matches. How many matches has he watched till now?

Error: 16 matches that Ram will see next month is irrelevant to the question being asked in the word problem.

- Error in entity/container/action Identification: Incorrectly identified entity/container is another issue that comes up. This usually occurs due to improper ellipsis or co-reference resolution.

  – शुरूआत में जेन के पास 87 केले थे। 7 1 घोड़े द्वारा खाए गए। अंत में जेन के पास कितने केले बचे?

  shurooaat mein jen ke paas 87 kele the. 7 1 ghode dvaara khae gae. ant mein jen ke paas kitane kele bache?

  Initially Jane had 87 bananas. 7 were eaten by 1 horse. How many bananas are left with Jane at the end?

  Error: 7 is not mapped to 'kele' by the solver and is therefore missed in calculation.

- Implicit Action/Entity/Container: This is a case when the action/entity/container is not explicit and the solver fails to understand it. An example of this error category is when the entity in question may not be the one required for answer.

  – 1 समुद्री जीवविज्ञानी ने 1 मछली मापी जो 0.3 फ़ीट लंबी थी और दूसरी मछली जो 0.2 फ़ीट लंबी थी। दूसरी मछली की तुलना में पहली मछली कितनी ज़्यादा लंबी थी?

  1 samudree jeevavigyaanee ne 1 machhalee maapee jo 0.3 feet lambee thee aur doosaree machhalee jo 0.2 feet lambee thee. doosaree machhalee kee tulana mein pahalee machhalee kitanee zyaada lambee thee?

  A marine biologist measured 1 fish that was 0.3 feet long and another that was 0.2 feet long. How much longer was the first fish than the second?

  Error: The solver identifies entities in this order 1 'samudree jeevavigyaanee' (marine biologist), 1 'machhalee' (fish), 0.3 feet, 0.2 feet. The entity in question is found to be 'machhalee' (fish), whereas the entity on which operation has to be performed is feet.

- Set Completion: The solver fails to handle word problems which require the knowledge of set completion. For example:

  – 4 बच्चों, 2 कर्मचारियों और 3 अध्यापकों का 1 समूह चिड़ियाघर जा रहा है। चिड़ियाघर कितने लोग जा रहे हैं?

  4 bachchon, 2 karmachaariyon aur 3 adhyaapakon ka 1 samooh chidiyaaghar ja raha hai. chidiyaaghar kitane log ja rahe hain?

  1 group consisting of 4 children, 2 staff and 3 teachers is going to zoo. How many people are going to the zoo?

  Error: Here, bacche (children), karmachaari (staff) and adhyaapak (teachers) form a set - log (people), which solver is not capable of identifying.

- Parsing Errors: Errors caused by incorrectly tagged part of speech. This also included cases when foreign words are missed by parsers. Examples of this error category are:
  - एवलिन के पास शुरुआत में 76 टॉफ़ियाँ थीं। क्रिस्टीन ने एवलिन को 72 टॉफ़ियाँ दीं। एवलिन के पास कितनी टॉफ़ियाँ हैं?

    evalin ke paas shuruaat mein 76 taaaifiyaan theen. kristeen ne evalin ko 72 taaaifiyaan deen. evalin ke paas kitanee taaaifiyaan hain?

    Evelyn initially had 76 candies. Christine gave 72 candies to Evelyn. How many candies does Evelyn have?

    Error: 'taaaifiyaan' gets tagged as VM i.e. verb in first statement.
  - मृदुल के MP3 प्लेयर पर 30 गाने थे। अगर उसने इसमें से 8 पुराने गाने डिलीट किए और फिर 10 नए गाने डाले, तो उसके MP3 प्लेयर पर कितने गाने हैं?

    mrdul ke mp3 pleyar par 30 gaane the. agar usane isamen se 8 puraane gaane **dileet** kie aur phir 10 nae gaane daale, to usake mp3 pleyar par kitane gaane hain?

    Mridul had 30 songs on his MP3 player. If he **deletes** 8 old songs from it and then adds 10 new songs, how many songs does he have on his MP3 player?

    Error: 'dileet' is identified as a foreign word and gets unknown label.

- Rules: There are cases when a rule that works for some examples may not work for others. Word problems that fall under this category are:
  - सिद्धांत के पास बैंक में 49 रुपये और 24 पैसे थीं। उसके पिता ने उसे 31 रुपये और 39 पैसे दिए। उसके पास अब कितने पैसे हैं?

    siddhaant ke paas baink mein 49 rupaye aur 24 paise theen. usake pita ne use 31 rupaye aur 39 paise die. usake paas ab kitane paise hain?

    Siddhant had 49 rupees and 24 paise in the bank. His father gave him 31 rupees and 39 paise. How much money does he have now?

    Error: When multiple monetary units are present they all get converted to the rupee symbol and lose the unit level of information. This is more prevalent in Hindi because the 'paise' is a unit as well as used to refer to the term 'money' in general. We need more granular rules for handling money.
  - 1 खिलौने वाले के पास 24 खिलौने थे। उसने 16 खिलौने बेच दिए। बताइए, अब उसके पास कितने खिलौने बचे?

    1 khilaune vaale ke paas 24 khilaune the. usane 16 khilaune bech die. bataie, ab usake paas kitane khilaune bache?

    1 toy seller had 24 toys. He sold 16 toys. How many toys does he have now?

    Error: When container is not a proper noun or place, the solver doesn't identify one at all.

- Others:
  - यदि अलमारी का मूल्य 1595 रुपये है और फ्रिज का मूल्य अलमारी के मूल्य से 6055 अधिक है तो अलमारी और फ्रिज का कुल मूल्य पता करो।

    yadi alamaaree ka mooly 1595 rupaye hai aur phrij ka mooly alamaaree ke mooly se 6055 adhik hai to alamaaree aur phrij ka kul mooly pata karo.

    If the cost of almirah is Rs 1595 and the cost of fridge is Rs 6055 more than the cost of almirah then find the total cost of almirah and fridge.

    Here, the solver does not understand that it has to first find the cost of 'phrij' (fridge) and then find the total.
  - फूलदान में 9 लाल गुलाब और 3 सफ़ेद गुलाब थे। सैली ने अपने बगीचे से 15 लाल गुलाब काटे और उन्हें फूलदान में रखा। फूलदान में अब कितने लाल गुलाब हैं?

    phooladaan mein 9 laal gulaab aur 3 safed gulaab the. sailee ne apane bageeche se 15 laal gulaab kaate aur unhen phooladaan mein rakha. phooladaan mein ab kitane laal gulaab hain?

    There were 9 red roses and 3 white roses in the vase. Sally cut 15 red roses from her garden and placed them in a vase. How many red roses are there in the vase now?

    Error: In this example, the states identified have entities as follows: 9 red roses, 3 white roses and -15 red roses because the verb category for 'kaate' (cut) is negative. Because of which 'rakha' (placed) also takes -15 and leads to an incorrect answer.

From these errors we understand that the solver can be improved by adding proper ellipsis/co-reference resolution, sentence simplification, adding more granular rules to handle more indicators, monetary units etc. This solver has another limitation that it calculates the answer but doesn't generate the equation, this way we are unable to directly analyse if the steps followed by the solver to reach the solution are correct or not.

5.4 Limitations of WPS using Verb Categorisation

Apart from the limitations of the solver, the method of using verb categorisation as a means to solve word problems also has some limitations. We have discussed these limitations in this section.

5.4.1 Limited to Addition and Subtraction

As stated in the beginning of this chapter, solving word problems using verb categorisation is only limited to Addition and Subtraction word problems because verbs can only help us identify these operations.

5.4.2 High Dependency on Parsers

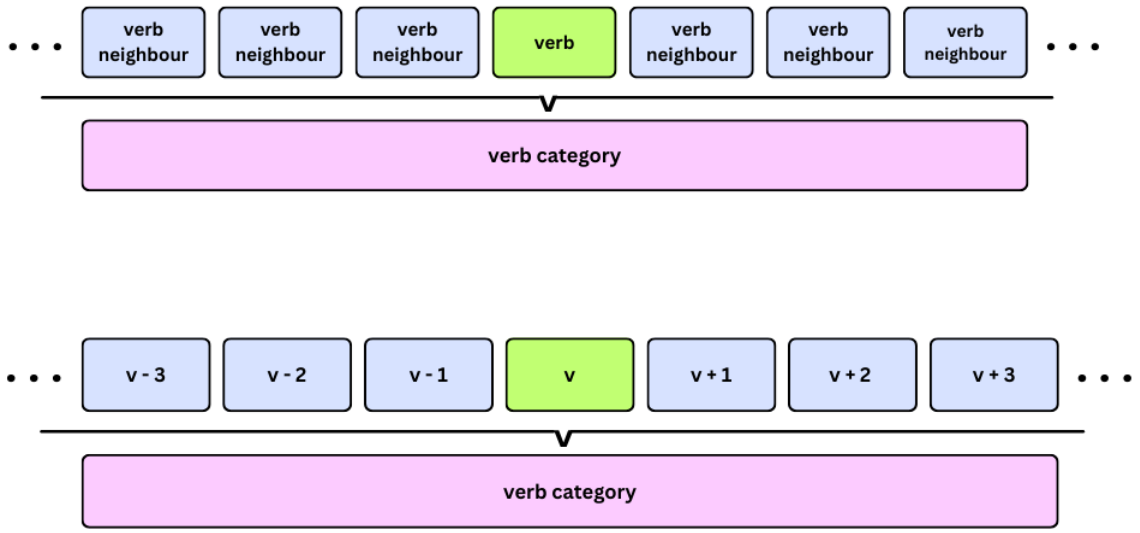
There were errors in the tags acquired from different Hindi parsers. Since verb categorisation very heavily relies on these parsers from finding verb categories to identifying entities, containers and actions/verbs for solving the word problems. This adds to the limitation of this method.

- **Errors in POS tags, UD tags by Parser.** While analysing the error caused due to parser, a significant number of errors were attributed to Parsing issues. Moreover, while performing verb categorisation too, non-verbs were tagged as verbs because of which we had to include the 'NA' category.

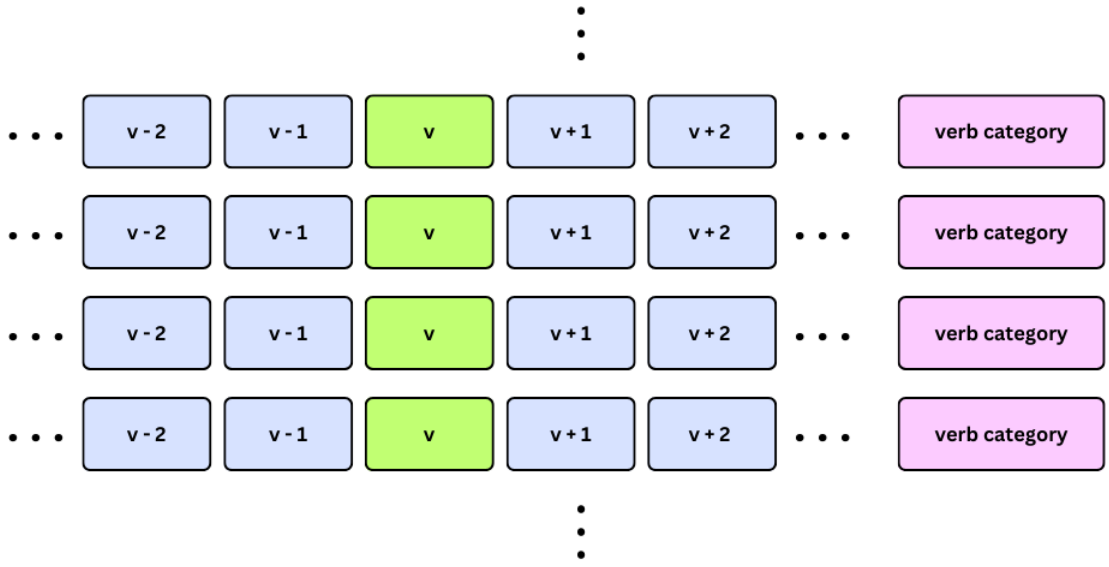
- **Errors in Tokenisation by Shallow Parser.** We also found some tokenisation errors. These errors increased even more for our tasks because we used Shallow Parser for generating root and POS tags of the words in the MWPs and ISCNLP parser for Universal Dependency(UD) tags. Both the parsers use their own tokenisers, therefore, two sets of token-tags were produced:

  - TokensI - Root, POS (from Shallow Parser)
  - TokensII - UD (from ISCNLP Parser)

  While mapping these two sets, some inconsistencies were noticed. These two sets of tokens i.e TokensI and TokensII did not match. Around 247 sentences were tokenised differently by the two parsers.



(a) One sample for WPS verb categorisation



(b) All samples for WPS verb categorisation

Figure 5.7: Desired data sample structure for WPS Verb Classification task

Chapter 6

Hindi Word Problem Solver: Deep learning Approaches for Word problem Solving

In this chapter we built an end-to-end Deep Learning based model to solve math word problems using the HAWP dataset. The motivation behind building this solver stems from the limited coverage of solving word problems using verb categorisation and high dependency on external tools as discussed in the previous chapter. For creating a wider and robust system, we propose this end-to-end deep learning based equation generator. This way, we are not only able to evaluate the model based on the answer it generates but using the equations it generates, we can understand how it reaches the solution.

We also propose the use and benefit of equation equivalence for evaluation of models that predict equations for word problems.

6.1 Setup

For this approach we pose word problem solving as sequence to sequence learning task. This task was implemented by using the open source open NMT [13] toolkit.

6.2 Preprocessing

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

- Word to Number Conversion
- Unit Conversion
- Special Number Token Replacement
- Equation Notation Conversion
- Conversion into Sub-words

Each of these steps is discussed in detail below.

6.2.1 Word to Number Conversion

Many a times numbers are written in 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. Now, in Hindi there are multiple relevant spellings of numbers in word form, for example, 27 is written as ‘ सत्तइस ’, ‘ सत्ताईस ’. Likewise, 28 is written as ‘ अठ्ठाइस ’, ‘ अट्ठाईस ’. We can observe this for many other numerals as well. We developed and used an in-house converter tool to perform this task.

6.2.2 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 6.1.

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 leetar
Time	2 ghante 45 minit	2.75 ghante

Table 6.1: Unit Conversion Examples

6.2.3 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 [31]. 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 [21]. 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 6.2: Implicit Quantity Examples

6.2.4 Equation Notation Conversion

All the equations are annotated in the infix notation. Many previous works [23, 7, 8] showed that deep neural architectures perform well when predicting equations in prefix notations. So all the infix notations were converted into corresponding prefix equations. Examples of conversion of infix to prefix expressions are given below:

Infix Annotated Equation	Equivalent Prefix Expressions
$X = (a + b) + c$	$X = +a + bc$
$X = a + (b - c)$	$X = +a - bc$
$X = a - (b + c)$	$X = -a + bc$

Table 6.3: Prefix Equivalents for Infix Expressions

6.2.5 Conversion into Sub-words

The final preprocessing step is the conversion of tokens into their subword forms. We used the BPEmb [1] package using pretrained subword embeddings and subword models to perform this task.

[1] `https://github.com/bheinzerling/bpemb`

6.3 Setting

We used the publicly available pre-trained subword embedding [9] for encoding our Hindi input data. These 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 [6] encoder decoder network with global attention [2] for predicting the prefix equations given a word problem. The hyper parameter details are shown in Table 6.4.

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 6.4: Configuration of BiLSTM model with Global Attention for Hindi

6.4 Evaluation

Most of the mathematical word problem solvers are evaluated either on equation accuracy or solution accuracy. For equation accuracy we use prefix expressions.

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 6.5 (examples include infix expressions for ease of understanding). We also observed that equation equivalence improves the performance of the model by 2% on an average.

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 6.5: Equation Equivalence Examples

6.5 Results and Error Analysis

We performed 10 fold cross validation on the whole dataset. The results are shown in table 6.6. Most of the errors are 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.

	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 6.6: Average Accuracy after 10-fold Cross Validation

If we analyse the second example in Table 6.7:

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 6.7: Examples of Erroneous Cases

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.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

To the best of our knowledge, the work described in this thesis is one of the first attempts to create a word problem solving dataset and solver for any Indian Language. In this thesis, we created a Hindi word problem solving dataset and developed baseline models for solving these Hindi word problems. Through such efforts, we hope to encourage research on word problem solving in languages other than English and Chinese by attempting to eliminate the problem of the unavailability of data.

Chapter 2 focuses on building a diverse and natural dataset for Hindi word problem solving. We start by looking at the structure of WPS datasets and the limitations of current datasets. We manually craft problems with the help of Math Textbooks and Math Teachers, along with a data augmentation technique to produce HAWP that consists of 2336 word problems in Hindi. The lexical diversity of the HAWP 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 annotate this data with equations, number of operations and relevant indices using our in-house equation annotation tool.

In Chapter 3, we dive deep into the augmentation process used, i.e. translation. We look at the various data augmentation techniques available and analyse their strengths and weaknesses. The common pain point for all augmentation techniques available is that they produce near duplicate data, which hampers the diversity of the dataset. This leads us to leverage the benchmark datasets available in English to augment our word problems. Next, we take the raw translation and define the steps we used to rectify and level up these translations so that they can be used to create a Hindi word problems dataset. For this purpose, we use techniques like localisation and borrowing. We also make changes to the translations to improve the naturalness and diversity of the dataset.

Chapter 4 serves as a conclusion for the process of dataset creation as we evaluate HAWP based on its previously claimed properties: naturalness and diversity. To determine how natural the dataset is, we asked students from Grades 6-7 to solve our word problems and found that approximately 90% of the word problems were solved correctly by these students. For determining the diversity of word problems, we use a number of diversity metrics proposed by different researchers. The results of this

exercise showed that HAWP had a lexical diversity comparable to other benchmark datasets available. This made us conclude that the HAWP dataset is natural and diverse.

We also developed baseline models for solving Hindi word problems. First of which is described in 5. This model uses verb categorisation to solve word problems. We categorise verbs into five different categories: observation, positive, negative, positive transfer and negative transfer. The verbs in HAWP are annotated with these categories. And this verb category annotated data is used to generate verb categories using three different approaches: 1) Closest verb model, which uses word vectors to find the nearest verb and use its category, 2) Context model, which uses a small context window to identify a verb's category and finally, 3) MuRIL contextual embeddings that use all words up till the verb in a sentence to determine verb category. Using these verb categories, we define a rule-based solver that generates the answer to the word problem with approximately 37% accuracy.

Due to the limitations of our model presented in Chapter 5, in Chapter 6 we developed a deep learning-based end-to-end solver that generates equations. In this chapter, we also proposed a new evaluation metric leveraging the equivalence property of mathematical equations.

With this, we conclude the thesis and hope our work will enthuse researchers towards this challenging NLP task in low-resource languages.

7.2 Future Work

7.2.1 Dataset

- Different data augmentation techniques can be explored to enhance the size of our dataset.
- More word problems with modulus operations and implicit quantities can be added to the dataset.

7.2.2 Solvers

- Our rule based solver can be improved to give improved results for addition and subtraction word problems by incorporating methodological improvements in ellipsis resolution, co-reference resolution.
- Moreover, the word problems can be simplified first to contain one verb per statement. This along with co-reference resolution will make it easier to identify entities, containers and verbs associated with each entity.
- For improving our end-to-end solver, the task of fine-tuning available BERT and other transformer based models can be taken up.

Related Publications

- **Harshita Sharma**, Pruthwik Mishra, and Dipti Sharma. 2022. **HAWP: a Dataset for Hindi Arithmetic Word Problem Solving**. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3479–3490, Marseille, France. European Language Resources Association

Bibliography

[1] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. 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 2019. Association for Computational Linguistics.

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

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

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

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

[7] K. Griffith and J. Kalita. 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, 2019.

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

[9] B. Heinzerling and M. Strube. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA).

[10] 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)*, pages 523–533, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[11] D. Huang, S. Shi, C.-Y. Lin, J. Yin, and W.-Y. Ma. 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, Aug. 2016. Association for Computational Linguistics.

[12] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*, 2021.

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

[14] R. Koncel-Kedziorski, H. Hajishirzi, A. Sabharwal, O. Etzioni, and S. D. Ang. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015.

[15] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, and H. Hajishirzi. 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, 2016.

[16] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay. 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, 2014.

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

[18] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[19] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[20] S.-y. Miao, C.-C. Liang, and K.-Y. Su. 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 2020. Association for Computational Linguistics.

[21] P. Mishra, L. J. Kurisinkel, D. M. Sharma, and V. Varma. 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 Dec. 2018. Association for Computational Linguistics.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 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 2002. Association for Computational Linguistics.

[23] A. Patel, S. Bhattamishra, and N. Goyal. 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 2021. Association for Computational Linguistics.

[24] S. Roy and D. Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[25] S. Roy and D. Roth. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*, 2016.

[26] S. Roy and D. Roth. Unit dependency graph and its application to arithmetic word problem solving. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31, 12 2016.

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

[28] 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*, pages 1132–1142, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[29] S. Upadhyay and M.-W. Chang. Draw: A challenging and diverse algebra word problem set. 2015.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Y. Wang, X. Liu, and S. Shi. 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, Sept. 2017. Association for Computational Linguistics.

[32] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification, 2016.

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