Word Problem Solving (WPS)

Dr. Pruthwik Mishra

AI Department, SVNIT Surat

December 21, 2024

Outline

Introduction

- Arithmetic word problem solving is a challenging task in NLP
- This involves solving a word problem automatically

Example

- Problem Text: "*India registered a hard-fought win against Australia. Indian won at the Chepauk by chasing down 199 in 41.2 overs. KL Rahul scored a match winning knock of 97 runs ably supported by the chase master Kohli's masterful 85. KL Rahul finished the game with a six over extra cover. India's final score was 201 for 4.*"
- Question: "*What was India's run rate?*"
- Question: "*How many runs did Indian batsmen, apart from Rahul and Kohli, score?*"

- The 1^{st} question involves a division operation evoked by the word rate.
  - What is the numerator and denominator?
  - What is the formula for calculating run rate?
  - Is any conversion required?

- The formula for 1^{st} word problem is:
  $\textit{Run Rate} = \textit{Total Runs Scored} \ / \ \textit{Total Overs}$
- Each over has 6 balls, to convert into a number conversion is required
  - 1 ball -> 1/6
  - 2 balls -> 2/6
  - ..
  - 6 balls -> 6/6
- Equation to Solve Question 1:
  $\textit{India's Run Rate} = 201 \ / \ 41.333$

- The 2^{nd} question involves addition and subtraction operations
- The formula for the 2^{nd} question is:
  $$\textit{Run Scored By Rest of the Indian Team} = \textit{Total Runs} - (\textit{Runs Made by Rahul} + \textit{Runs Made by Kohli} + \textit{Extras})$$
- This question cannot be answered as the information about the *Extras* is not present in the given text

Research Questions

- Not all the sentences in a word problem is relevant
- Identification of relevant components to solve the word problem
    - Operands
    - Operations

- Automatic math solvers can assist both students and teachers
  - Steps required to solve a word problem
    - Understanding the problem (involves NLU)
    - Forming the equations
    - Solving the equations
  - Word Problem Solving (WPS) is a complex NLU task
  - Natural Language Understanding (NLU) is the most challenging aspect of NLP

Our Approaches and Our Contribution

- Our Approaches
  - Frame Based Solvers
  - Neural Solvers
- Our Contribution
  - Development of Solvers
  - Initial models were developed for English
  - Extended the models to Indian Languages

Frame Based Word Problem Solver

- Consists of 2 steps
  - Frame Identification
  - Rule based solver on identified frames to create equations and solve them
- We explored this approach initially since math word problems largely hinge on verbs

Introduction of Frames

- **Frame**: A basic computational unit consisting of relevant information for solving a word problem
- Categorize each frame into *Action Frame* and *State Frame*

Slots inside Frames

- **Slot**: A unique identifier for a frame
- *Key Assumption*: Each sentence consists of a single verb
- Slots are filled from the dependency parse of a sentence
- What are the slots?
  - Entity Holder
  - Entity
  - Quantity of Entity
  - Recipient (For transfer verbs)
  - Additional Information (such as time, place etc.)

Frame Dataset

- For identifying frames, we need an annotated dataset with frame labels
- As frame annotated data was not available, we manually annotated frames in word problems
- Frames are identified by the verbs and their associated context
- Total types of frames=22
- The questions for annotation are selected from the worksheets available under `https://www.math-aids.com/Word_Problems/`
- To facilitate annotation, we develop an offline frame annotation tool

Annotated Frame Dataset Details

The annotated dataset is split into *Train* and *Test*

Type	**#Questions**	**#Sentences**	**#Frames**
Train	444	1253	1253
Test	60	168	168
Total	504	1421	1421

Table: Annotated Corpus Size

Approaches for Frame Identification

- Machine Learning Approaches
  - Each sentence is represented as a TF-IDF vector
  - Experimented with both word and character unigrams
  - Implemented SVM and Random forest classifiers for it
- Deep Neural Approaches
  - Distil-RoBERTa-base: 82 million parameters
  - RoBERTa-base: 125 million parameters

Frame Identification Results

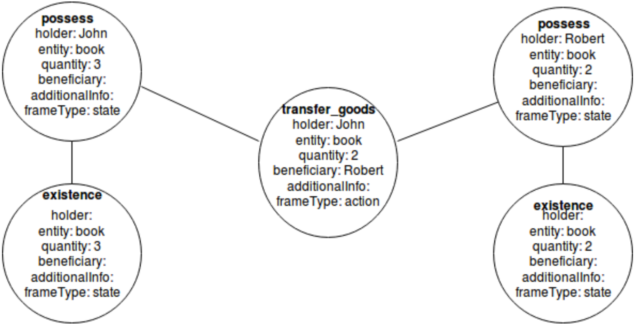
Evaluation done using 5-Fold stratified cross validation

Model	Features	F1-Score
Linear SVM	Word uni+char[3,6]	0.88
Random-Forest	Word uni+char[3,6]	0.86
Distil-RoBERTa-base		0.94
RoBERTa-base		**0.95**

Table: Frame Classification Results for Different Approaches

Working Example

John had 3 books. He gave 2 to Robert. How many books does John have now?



Results for Solver

- Tested on 302 word problems in AI2 [1] benchmark dataset consisting of single addition and subtraction operations
- Used the best performing frame identifier (see Table 2)
- The frame based solver solved 43.2% (130 out of 302)

Error Analysis

Error Type	Example	Explanation
World Knowledge	Students at Arcadia schools are participating in a coat drive. 9437 coats have been collected so far. 6922 coats were collected from the high schools, and the ***rest*** from the elementary schools. How many coats were collected at the elementary schools?	The word ***rest*** denotes a negative operation. Our frame identification system depends on a verb for identifying an operation. In this case, rest is not a verb.
Incorrect Frame Identification	Mary is baking a cake. The recipe calls for 7 cups of flour and 3 cups of sugar. She already ***put*** in 2 cups of flour. How many cups of flour does she need to add?	Here put is identified as add_to_group frame which invokes an addition operation. Here in this problem, it is associated with a subtraction operation.
Non-Linear Sequence	Joan found 70 seashells on the beach. ***Joan gave Sam some of her seashells.*** She has 27 seashells. How many seashells did she give to Sam?	The system expects the sentences/events to be linear. This word problem can be formulated as: Initial + Change = Final If Initial and Change are given, our system computes the Final. In this example, Initial and Final are given, Change is asked. This reordering breaks the system.

Figure: Error Analysis of the Frame Based Solver

Coreference	Joan grew 24 pumpkins, Keith grew 42 pumpkins, and Alyssa grew 13 pumpkins. ***They*** worked for 34 days on the farm. How many pumpkins did they grow in all?	Stanford Coreference tool assigns the coreferent of ***They*** as ***24 pumpkins***.
Parsing Errors	Tim had 7 quarters and 9 nickels in his bank. His dad gave him 3 nickels and 5 pennies. How many ***nickels*** does Tim have ***now***?	Our system tries to identify the frame and its slots asked in the question sentence. Here, the stanford dependency parser marks ***now*** as the direct object or the entity and ***nickels*** as the adjectival modifier. As the slots are incorrectly identified, the system fails to solve the word problem.

Figure: Error Analysis of the Frame Based Solver

Discussion

- Automatic solvers need to be robust and should be able to solve a variety of word problems
- Although frame based solver provided well explainable solutions, they failed to solve majority of the word problems
- As neural methods have shown significant improvement in the performance of many NLU tasks, we lean towards developing neural word problem solvers

Deep Learning Based Solvers

Overview

- Two Approaches
  - Approach 1: Decompose equation generation into two subtasks
    1. Operand Identification
    2. Operation Identification
  - Approach 2: End-to-End Equation Generation using different variants of encoder-decoder architecture
  - Models are trained on English *addition* and *subtraction* word problems

Example

- Question: *Ramesh had 10 poetry books and 10 novels. He gave 5 novels to Suresh. How many novels does Ramesh have now?*
- We will use this example in the coming slides.

Decomposition of Equation Generation

Operand Identification

- Similarity Matching [training-free classifier]
    - Take the context around a quantity or number
    - Find the similarity with the question sentence
    - Different context window sizes (1, 2, 3)
- Operand Relevance Prediction as a classification task by fine-tuning pre-trained transformer models

Operation Identification

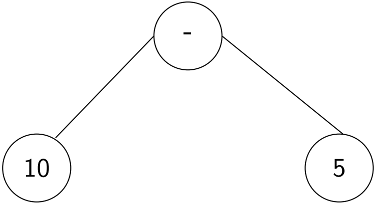
- BiLSTM
  - Split the word problem into two parts: problem text and question sentence
  - Learn representations for both separately
  - Concatenate the two representations
  - Apply a softmax layer to predict the operation
- Fine-tuning pre-trained transformer models

Dataset

- Used publicly available MAWPS [2] dataset
- Contains 1751 single step arithmetic word problems involving a single addition or subtraction operation
- Annotated data with quantity relevance was not publicly available
- So, we annotated 3718 such samples from 1751 sentences
- 2 settings
  - Tokens in a context window around a quantity
  - Tokens in a context window around a quantity and the question sentence
  - Size of context window=7

Equation Formation

- An equation is formed by the composition of the results from the two tasks
- Let us take the example 25
- Relevant Operands: 10 novels, 5 novels
- Operation: Subtraction (-)
- Equation: $x = 10 - 5$
- We have used Equation Accuracy to evaluate solvers



Equation Accuracies

Model	Config	Operand	Operation	Equation
DILTON	CW=1	92.25	88.01	81.5
Distilroberta-base	-Question	90.32	94.06	89.3
RoBERTa-base		90.76	**95.89**	90.9
Distilroberta-base	+Question	91.5	94.06	89.8
RoBERTa-base		92.1	**95.89**	**91.3**

Table: 5 Fold Equation Accuracies by Composing Operand and Operation Prediction

Discussion

- BERT based models outperform other approaches
- Distilled versions are also comparable
- Errors in operation prediction due to the ambiguity between similar operations
- Unable to handle word problems with more than one operators
- Therefore, we design end-to-end systems capable of generating equations at once

End-to-End Equation Generation

Introduction

- Modeled as a sequence to sequence learning task
- Input is the word problem (or word problem + Question), Output is the required equation to solve the problem
- Question: *Ramesh had 10 poetry books and 10 novels. He gave 5 novels to Suresh. How many novels does Ramesh have now?*
- Equation: x=10-5
- 3 approaches
  - Memory Network based Encoder and LSTM based Decoder [EquGener]
  - BiLSTM based Encoder and Decoder
  - Transformer based Encoder and Decoder

Architecture

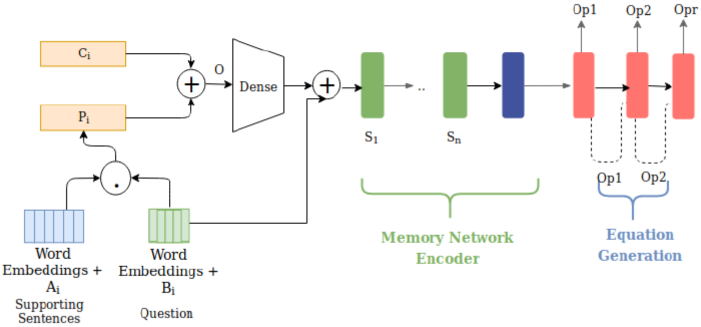


Figure: Architecture of EquGener

Dataset

- Split the data into *Train* and *Test*
- The Test data is divided into 3 parts: MA1(all relevant), MA2(more irrelevant info), IXL(more information gap)

Type	**Size**
Train	1314
MA1	103
MA2	118
IXL	81

Table: Dataset Details for End-to-End Equation Generation

Results on AI2 Dataset

System	MA1	IXL	MA2
EquGener	94.18	85.19	55.08
BiLSTM with Attention	94.18	90.12	87.29
T5-Large	**97.09**	**97.53**	**96.61**

Table: Comparison of results on AI2 dataset for different models. Numerical values represent % of problems solved

Benchmark Datasets and Their Characteristics

Benchmark Datasets

- We need benchmark datasets to train and test word problem solvers
- Most are developed by crawling popular math tutoring websites such as `www.algebra.com`, `www.mathtutor.com`, `www.math-aids.com`
- **AI2** [1]: a curated version of such problems with a very limited vocabulary
- **MAWPS** [2]: This study showed that many benchmark datasets were composed of problems with high lexical overlap and equation template overlap.
- **ALGES514**: [3] contains 514 problems from only 28 equation templates
- **Unbiased** [5]: Removed existing biases from existing benchmark corpora and create a dataset comprising of 1492 word problems

Issues in Benchmark Datasets

- Issues
  - Lexical Overlap
    1. *Joan went to 4 football games this year . She went to 9 games last year . How many football games did Joan go to in all ?*
    2. *John went to 5 football games this year . He went to 6 games last year . How many football games did John go to in all ?*
  - Ungrammaticality
    - *Joan found 70 seashells on the beach . she gave Sam some of her seashells . She has 27 seashell . How many seashells did she give to Sam ?*

Can we resolve this issue?

Technique to reduce overlap

- Developed a technique to remove highly overlapping word problems in any dataset
- Let D denote a dataset containing n word problems.
  $LexSim(p, q) \; = \; |T(p) \cap T(q)|/|T(p) \cup T(q)|$
  For any word problem $p_i \in D$, remove all the word problems from D satisfying the property $LexSim(p_i, p_j) >= th$

Lexical Overlap Reduction in Benchmark Datasets

Dataset	Size	Reduced Size For Different Similarity Thresholds					
		0.5	0.6	0.7	0.8	0.9	1.0
AI2	395	185	228	275	333	380	389
ASDiv	2305	1948	2131	2227	2274	2298	2298
IL	562	269	333	394	444	481	483
Single-Eq	508	353	386	416	449	483	496
MAWPS	2373	894	1035	1179	1316	1450	1802
CC	600	114	116	117	118	121	364
Unbiased	1492	856	1035	1197	1327	1431	1473

Table: Reduction of Datasizes after removal of similar problems

With so less data to train models, can we increase the data size?

Data Augmentation (DA)

Augmentation Strategies

- 2 kinds of augmentation
  - Lexical Augmentation/Augmentation Using POS Tagging and Paraphrase Tables
  - Augmentation Using Translation

Augmentation Using POS Tagging and Paraphrase Tables

- Bin the content words present in the paraphrase table into 4 classes: Adjective, Adverb, Noun, Verb
- Selection based on Distributional Similarity
- Additional Constraints
  - Synonym Similarity
  - Language Model
  - Gazetteer List

Example of Generation

- Original Word Problem: *For Halloween Debby and her sister combined the candy they received . Debby had 32 pieces of candy while her sister had 42 . If they ate 35 pieces the first night , how many pieces do they have left ?*
- Generated Word Problems:
  1. *For <u>christmas</u> Debby and her <u>daughter</u> combined the <u>lollipop</u> they received . Debby had 32 <u>artifacts</u> of <u>lollipop</u> while her <u>daughter</u> had 42 . If they ate 35 <u>artifacts</u> the first <u>evening</u> , how many <u>artifacts</u> do they have left ?* [n=3]
  2. *For <u>birthday</u> Debby and her <u>grandmother</u> combined the <u>lollipop</u> they received . Debby had 32 <u>artifacts</u> of <u>lollipop</u> while her <u>grandmother</u> had 42 . If they ate 35 <u>artifacts</u> the first <u>hours</u> , how many <u>artifacts</u> do they have left ?* [n=5]

Augmentation Using Translation

- The previous approach can only generate word problems that are structurally very similar without much diversity
- Translating a diverse dataset can solve this issue
- Ape210K is the largest word problem dataset with 210488 word problems and 56532 equation templates; but available in Chinese
- 123430 word problems were translated into English using Google Translate API

Noise Removal from Translations

- Named Entity Matching
- Non-ASCII Characters
- Erroneous or missing text
- Sample Validation

Does data augmentation help improve a solver's performance?

Impact of Data Augmentation

- Experiments were done to verify how diversity and equation notations impact performance of data augmentation.
- 2 datasets were chosen.
  - AI2 (low diversity)
  - ASDiv (high diversity)
- Equation Notations
  - Infix: $(a \ + \ b)$
  - Prefix: $+ \ a \ b$
  - Postfix: $a \ b +$

DA on AI2 dataset

Dataset	Train_Samples	Validation_Samples	Total
AI2	316	79	395
AI2+Augment	2072	517	2589

Table: Corpus Details for Data Augmentation Experiments

Model	Equation_Accuracy
AI2 Model	34.268
AI2+Augment Model	42.368

Table: Results of Solvers with Augmentation on IL dataset

ASDiv dataset

ASDiv dataset (Academia Sinica Diverse MWP Dataset) is diverse dataset.

No_of_Operation	#Word_Problems
1	985
2	338
Total	**1323**

Table: Distribution of Chosen ASDiv Word Problems

DA on ASDiv dataset with Paraphrase Tables

No_of_Operation	#Word_Problems
1	1993
2	930
Total	**2923**

Table: Distribution of Augmented Word Problems with Original

No_of_Operation	#Word_Problems
1	4952
2	5613
3	1476
4	530
Total	**12571**

Table: Distribution of Operations for the Full Augmentated Dataset

Does number replacement strategy impact a solver's performance?

Results on Non-Augmented Dataset

- 2 types of symbol replacement: $(p + q)$ Vs $(number0 + number1)$

Config	Notation	Equation_Accuracy
Single Letters	Infix	43.91
	Postfix	45.04
	Prefix	44.06
String with Occurrence count	Infix	**45.5**
	Postfix	**48.29**
	Prefix	**49.96**

Table: 5 Fold Results With Different Number Representations on Original ASDiv Dataset

Evaluation Metrics

- As explained earlier, we have used *Equation Accuracy* to evaluate our solvers
- Additionally, we have used the concept of *Equation Equivalence*
- We propose an evaluation metric of *Equation Accuracies with Equivalence*

Results on Augmented Datasets

For all the experiments on the English augmented datasets, we used the pre-trained XLM RoBERTa-base which has 270 million parameters.

Config	Notation	Equation_Accuracy
Lexical Aug+Orig	Infix	49.1
	Postfix	49.9
	Prefix	47.6
Lexical Aug+Translation +Orig	Infix	46.4
	Postfix	48.1
	Prefix	46.6
Lexical Aug+Translation +Orig+1 and 2 operations	Infix	**53.7**
	Postfix	**56.5**
	Prefix	**54.6**

Table: Exact Equation Accuracies in the Full Augmentation Dataset

Equivalent Equation Accuracies

Config	Notation	Equation_Accuracy
Lexical Aug+Orig	Infix	50.3
	Postfix	50.7
	Prefix	50.6
Lexical Aug+Translation +Orig	Infix	50.0
	Postfix	50.3
	Prefix	47.7
Lexical Aug+Translation +Orig+1 and 2 operations	Infix	**56.7**
	Postfix	**58.4**
	Prefix	**56.0**

Table: Equation Accuracies with Equivalence in the Full Augmentation Dataset

Does sampling impact a solver's performance?

Sampled Evaluation

We randomly sampled 1600 translations from all the full translations keeping the same distribution of operations as given in table 10.

No_of_Operation	#Word_Problems
1	3001
2	1522
Total	**4523**

Table: Distribution of Sampled Augmentated Dataset

Results on Sampled Set

Config	Notation	Equation_Accuracy
Lexical Aug+Orig	Infix	49.12
	Postfix	49.88
	Prefix	47.6
Translation+Orig	Infix	45.5
	Postfix	48.29
	Prefix	49.96
Lexical Aug+Translation +Orig	Infix	**55.61**
	Postfix	53.97
	Prefix	54.86

Table: 5 Fold Results With Sampled Augmented Datasets with Exact Equation Accuracy

Discussion

- No significant difference in results between the full and sampled datasets
- Increase in data size is not directly proportional to performance
- Scores of augmentation with translation is poorer than scores of lexical augmentation

Does diversity play a role in a solver's performance?

Corpus Lexicon Diversity

- Corpus Lexicon Diversity (CLD) for any corpus measures the lexical diversity of a constituent word problem with other word problems in a corpus.
- Compute the similarity scores between two word problem using the BLEU
- Meta symbols are used
  - Original Text: *Ellen has 6 more balls than Marin. Marin has 9 balls. How many balls does Ellen have?*
  - Normalized Text: *<u>PERSON</u> has <u>NUMBER</u> more balls than <u>PERSON</u>. <u>PERSON</u> has <u>NUMBER</u> balls. How many balls does <u>PERSON</u> have?*

CLD for Datasets

Dataset	CLD
ASDiv	0.71
ASDiv + Lexical Aug	0.32
ASDiv + Translation	0.63
ASDiv + Lexical Aug + Sampled Translation	0.41
ASDiv + Lexical Aug + Full Translation	0.44

Table: CLD Scores of Different Datasets

Effects of Diversity on Solvers

- Solvers trained and tested on lexically similar word problems report high accuracies without generalizing well
- Data augmentation improves the performance of a solver
- Lexical augmentation paired with translation provides the maximum boost to a solver

WPS in Indian Languages

Introduction

- Unlike English, Chinese, very few attempts are made to develop solvers in Indian Languages (ILs)
- A dataset HAWP for solving word problems is available in Hindi consisting of 2336 word problems
  - Hindi Example: कोयल के पास 100 रुपये का 1 नोट है । दुकानदार इस नोट के बदले 20 रुपये के कितने नोट देगा ?
  - English Gloss: koyal ke paas 100 rupaye ka 1 not hai . dukaanadaar is not ke badale 20 rupaye ke kitane not dega ?
- No datasets for other Indian languages are available

Lexical Augmentation

- Can not replicate the same method as English because of unavailability of Hindi paraphrase tables
- But augmentation with word2vec pre-trained embeddings, gazetteers list, and language model is possible
- Also imposed a constraint on maximum possible generation

Example

- Original: शिखा ने 1 दुकान से 65 रु. का सामान खरीदा । उसने दुकानदार को 100 रु . का नोट दिया । बताओ , उसे कितने रुपये वापिस मिले ?
- English Translation: Sikha bought goods of worth 65 rupees from a shop. She gave a 100 rupee note to the shopkeeper. Tell, how much money will she get in return?
- Gloss in Roman: shikha ne 1 dukaan se 65 ru . ka saamaan khareeda . usane dukaanadaar ko 100 ru . ka not diya . batao , use kitane rupaye vaapis mile ?
- Generated: सलमा ने 1 दूकान से 65 रूपए . का बैग ख़रीदा । उसने व्यापारी को 100 रूपए . का नोट दिया । बताओ , उसे कितने रुपये वापिस मिले ?
- Gloss in Roman: salama ne 1 dookaan se 65 roope . ka baig khareeda . usane vyaapaaree ko 100 roope . ka not diya . batao , use kitane rupaye vaapis mile ?

Dataset Details for Lexical Augmentation Experiments

- HAWP consists of 2336 word problems
- CLD score of HAWP = 0.65
- This dataset contains single and two operation word problems in a ratio of 3: 1
- For creating a more balanced dataset, we undersampled this to maintain a ratio of 2: 1

config	No_of_Operation	#Word_Problems
No Augmentation	1	860
	2	430
	Total	**1290**
With Augmentation	1	3318
	2	2108
	Total	**5426**

Results on Augmented Dataset in Hindi

5 fold cross validation was done to evaluate the solvers.

Config	Notation	Equation_Accuracy	CLD
No Augmentation	Infix	34.81	0.648
	Postfix	38.53	
	Prefix	35.64	
With Augmentation	Infix	42.06	0.194
	Postfix	43.39	
	Prefix	44.57	

Table: 5 Fold Results For Hindi

Augmentation with Translation

- Similar to English, we augmented Indian language datasets with translations
- Except Hindi, word problem datasets for other Indian languages are not available
- Two languages are chosen for this task
  - Hindi - Resource Available
  - Telugu - No resource Available
- For creating datasets in these two languages, English translations of Ape210K are then translated into Hindi and Telugu using [4] transformer based MT systems

Is there a need to create a benchmark dataset?

Benchmark Datasets and their Creation

- Quality benchmark datasets are crucial for testing robustness of developed models
- In this work, we release a 3-way parallel word problem datasets in English, Hindi, and Telugu created manually containing 1127 word problems
- The problems were chosen from the English translations of the validation set of the Ape210K dataset and transcreated with post-editing
- Methods of Creation
  - Correct the equations if erroneous
  - If a word problem is illegible, then form a new word problem taking the equation as a basis
  - The verified and corrected word problems are then translated into Hindi and Telugu manually
  - For each language, two experts with post graduation level of education were selected
  - Translators have more than 5 years' of experience in translation

English Word Problem Transcreation with Post-editing

- Ensuring Grammatical Correctness
- Naturalness
- Localisation
- Simplification

Translated Word Problems

- Adequate
- Fluent
- Natural
- Transliteration of mathematical terms

Datasets Used

Language	#Train	#Validation	#Test
Hindi	13242	1472	1127
Telugu	8379	932	1127

Table: Dataset Details for Full Augmentation in Hindi and Telugu

Approaches

- Our English word problem solver is finetuned on XLM RoBERTa base
- Used multilingual XLM RoBERTa base for our experiments which supports Hindi, and Telugu
- Initial experiments explored the zero shot capabilities of our English XLM RoBERTa model
- We fine tune Hindi and Telugu solver on top of the English solver

Zero Shot Results

Lang	**Type**	**Infix**		**Postfix**		**Prefix**	
		Full	**1+2op**	**Full**	**1+2op**	**Full**	**1+2op**
Hindi	Val	26.4	30.2	30.6	35.4	24.8	28.8
	Test	19.2	26.0	19.7	26.8	18.2	24.6
Telugu	Val	23.4	33.1	19.8	28.5	20.2	28.8
	Test	20.1	26.8	13.7	18.3	19.0	25.7

Table: Zero Shot Experiment Results for Hindi and Telugu

Exact Equation Accuracies of IL Solvers

Lang	Type	Infix		Postfix		Prefix	
		Full	1+2op	Full	1+2op	Full	1+2op
Hindi	Val	55.2	60.9	51.8	58.0	52.4	58.9
	Test	26.5	32.1	25.1	31.5	23.0	28.9
Telugu	Val	39.5	47.3	35.1	43.8	32.9	41.1
	Test	24.2	29.8	21.6	26.6	20.4	27.3

Table: Exact Equation Accuracies on English Fine Tuned Model for Hindi and Telugu

Equaivalent Equation Accuracies of IL Solvers

Lang	Type	Infix		Postfix		Prefix	
		Full	1+2op	Full	1+2op	Full	1+2op
Hindi	Val	58.4	64.1	54.5	60.8	54.6	61.4
	Test	38.2	47.6	35.01	45.1	30.8	40.1
Telugu	Val	50.0	56.7	50.3	58.5	47.7	56.0
	Test	33.4	42.4	31.8	41.0	26.3	36.7

Table: Equivalent Equation Accuracies on English Fine Tuned Model for Hindi and Telugu

Discussion

- Fine tuning a pre-trained model is the way forward
- Augmentation improves a solver's performance if human created word problems are available
- With increase in number of operations the solver's performance gradually drops
- Equation equivalence acts as an equalizer

Word Problem Solving Capabilities of ChatGPT

- Tested on 10 problems each
- ChatGPT performs well in English
- ChatGPT performs poorly for Indian languages
- The major errors can be attributed to incorrect operand, operator identification and incorrect equation usage

Use of WPS Components in NLP Applications

Overview

- Errors in translating numbers when written in words
  - Developed Word2Number converters for English and Indian Languages
- Equation Identification and Conversion into Math Notations in Transcripts
  - A transcript is the textual form of what is spoken in an audio
  - Contains plain text even if the speech is about a mathematical concept
  - Developed an equation identifier and converter for English

Conclusion

- Experimented with different word problem solving approaches
- Showed the efficacy of deep learning models and their limitations
- Thorough experiments on the generation of different equation notations
- Developed of a new evaluation method using equation equivalence
- Developed of a new method to reduce overlapping between word problems in a dataset
- Developed of word problem solvers for Hindi and Telugu
- Created benchmark datasets for English, Hindi, and Telugu
- Developed equation identifiers and converted for English speech transcripts

Related Publications

- Purvanshi Mehta, Pruthwik Mishra, Vinayak Athavale, Manish Shrivastava, and Dipti Misra Sharma, "Deep Neural Network based system for solving Arithmetic Word problems", Proceedings of the IJCNLP 2017, System Demonstrations, Tapei, Taiwan, pages 65-68
- Pruthwik Mishra, Litton J Kurisinkel, and Dipti Misra Sharma, "Arithmetic Word Problem Solver using Frame Identification", CICLing-2018
- Pruthwik Mishra, Litton J Kurisinkel, Dipti Misra Sharma, and Vasudeva Varma, "EquGener: A Reasoning Network for Word Problem Solving by Generating Arithmetic Equations", Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, Hong Kong, pages 456-465

- Vandan Mujadia, Pruthwik Mishra, and Dipti Misra Sharma, "Deep Contextual Punctuator for NLG Text", SEPP-NLG 2021 Shared Task
- Harshita Sharma, Pruthwik Mishra, and Dipti Misra Sharma, "HAWP: a Dataset for Hindi Arithmetic Word Problem Solving", Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, pages 3479–3490
- Pruthwik Mishra, Litton J Kurisinkel, and Dipti Misra Sharma, "Arithmetic Word Problem Solver using Frame Identification", Extended Version, POLIBITS Journal, Not Published

- Pruthwik Mishra, Vandan Mujadia, and Dipti Misra Sharma, "Multi Task Learning Based Shallow Parsing for Indian Languages", TALLIP Journal, 2024
- Harshita Sharma, Pruthwik Mishra, and Dipti Misra Sharma, "Verb Categorisation for Hindi Word Problem Solving", ICON-2023

Questions?

References I

- Hosseini, M.J., Hajishirzi, H., Etzioni, O., Kushman, N.: Learning to solve arithmetic word problems with verb categorization. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 523–533 (2014)
- Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., Hajishirzi, H.: 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. pp. 1152–1157 (2016)

References II

- Kushman, N., Artzi, Y., Zettlemoyer, L., Barzilay, R.: Learning to automatically solve algebra word problems. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 271–281 (2014)
- Mujadia, V., Sharma, D.M.: Boosting mt performance for indian languages (2022), to appear
- Roy, S., Roth, D.: Mapping to declarative knowledge for word problem solving. Transactions of the Association for Computational Linguistics 6, 159–172 (2018)