Math Word Problem Generation with Multilingual Language Models

Kashyapa Niyarepola,Dineth Athapaththu,Savindu Ekanayake,Surangika Ranathunga
Department of Computer Science and Engineering, University of Moratuwa
Katubedda 10400, Sri Lanka
[kashyapabandara.17, dinethnaradaam.17,
savinduekanayake.17, surangika]@cse.mrt.ac.lk

Abstract

Auto regressive text generation for low-resource languages, particularly the option of using pre-trained language models, is a relatively under-explored problem. In this paper, we model Math Word Problem (MWP) generation as an auto-regressive text generation problem. We evaluate the pre-trained sequence-to-sequence language models (mBART and mT5) in the context of two low-resource languages, Sinhala and Tamil, as well as English. For the evaluation, we create a multi-way parallel MWP dataset for the considered languages. Our empirical evaluation analyses how the performance of the pre-trained models is affected by the (1) amount of language data used during pre-training, (2) amount of data used in fine-tuning, (3) input seed length and (4) context differences in MWPs. Our results reveal that the considered pre-trained models are capable of generating meaningful MWPs even for the languages under-represented in these models, even though the amount of fine-tuning data and seed length are small. Our human evaluation shows that a Mathematics tutor can edit a generation question fairly easily, thus highlighting the practical utility ofautomatically generating MWPs.

1 Introduction

Despite being one of the most important subjects, many school children find Mathematics difficult (Acharya, 2017), with many exams reporting high failure rates in Mathematics (Rylands and Coady, 2009). One way of improving Mathematics skills is to practice solving Mathematics problems (Thompson, 1985). However, this places extra burden on the tutors - they have to create different Mathematics questions and grade student answers. The alternative is to automatically generate Mathematics questions and grade student answers. The need of such systems that support as many languages as possible, is even more pronounced during the times of pandemics and war, where students get confined to homes/shelters without access to physical schools.

In this paper, we focus on the problem of automatically generating Mathematical Word problems (MWPs). Considering the fact that learning Mathematics is not a privilege to students speaking a particular language, we want to investigate the possibility of MWP generation in multiple languages. An MWP is a "narrative with a specific topic that provides clues to the correct equation with numerical quantities and variables therein" (Zhou and Huang, 2019). MWPs can be in categories such as algebra, geometry and statistics. An elementary MWP written in English is shown in the below example.

Rosy made cookies and she used 2 kg flour and 1.5 kg sugar. How much more flour than sugar did Rosy use?

Early solutions to MWP generation relied on template-based approaches (Polozov et al., 2015), and question rewriting (Koncel-Kedziorski et al., 2016). More recently, Recurrent Neural Networks (RNN) (Zhou and Huang, 2019; Liyanage and Ranathunga, 2020), fine-tuning pre-trained language models (Wang et al., 2021) as well as Variational Autoencoders (VAE) (Liu et al., 2020; Cao et al., 2021) have been proposed. Only Liyanage and Ranathunga (2020) have tried their NN solution in a multilingual setting, however the results are sub-optimal.

Thus, our objective is to investigate the use of multilingual pre-trained models for MWP generation. Here, we treat MWP generation as an auto-regressive problem - the system has to generate a question starting with the provided seed (the starting portion of the question that is expected to be generated). Compared to text generation tasks such as story generation (Roemmele, 2016) or news generation (Leppänen et al., 2017), MWP generation is challenging because MWPs have mathematical constraints, units and numerical values as shown in the above example.

As mentioned above, auto-regressive language models such as GPT-x (Radford et al., 2019) have been already used for MWP generation (Wang et al., 2021). They are a common choice for Natural Language Generation (NLG) tasks (Lee and Hsiang, 2020; Mosallanezhad et al., 2020; Budzianowski and Vulić, 2019). Sequence-to-sequence models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2019) have also been used for NLG in an auto-regressive manner (Tan et al., 2020; Lewis et al., 2020). However, this option has been used to a lesser extent compared to GPT-x in similar text generation tasks, and never for MWP generation.

Despite their success on English text generation, GPT-x models are not available for other languages. Building multilingual or language-specific GPT models is not practical for many languages, particularly the low-resource ones. In contrast, T5 and BART both have their multilingual versions: mT5 (Xue et al., 2020) and mBART (Tang et al., 2020) (respectively). We are only aware of the empirical analysis of Chen et al. (2021), who tested the auto-regressive text generation capabilities of mT5 and mBART in the context of 4 high resource languages (for four tasks: story, question and title generation).

We carry out an empirical study on the mBART and mT5 models for MWP generation, considering two low-resource languages Sinhala and Tamil, along with English. All these languages are included in mBART and mT5. For a more comprehensive analysis, we evaluate T5, BART and GPT-2 for English MWP generation as well. Our experiments answer four important questions:

1. How the performance of mT5 and mBART varies depending on the language - because, for the related Machine Translation task, it has been shown that the model performance on individual languages depends on the amount of language-specific data used during model pre-training (Lee et al., 2022)

2. How the performance of the models varies depending on the amount of fine-tuning data - because for many languages, having a large training set is not realistic

3. How much information (size of the seed) should be provided to the model at the inference stage for it to generate a meaningful MWP - because a tutor should be able to generate a new MWP by providing minimal information.

4. How the context of an MWP affects the generation performance - because there is a wide variety of MWPs

As an additional contribution, we create a benchmark dataset by extending the dataset created by Liyanage and Ranathunga (2020) for MWP generation. Each English question was manually translated to Sinhala and Tamil, creating a multi-way parallel dataset. Our dataset is publicly released[1], and can be considered as a test set even for Machine Translation.

We believe that our work is the first to conduct an empirical analysis on the use of (1) GPT, BART, T5, mBART and mT5 for auto-regressive generation of MWPs and (2) mBART and mT5 for the general task of auto-regressive text generation considering low-resource languages. Our findings are indeed very promising for low-resource languages. Even for very small seeds and fine-tuning dataset sizes, these models (mBART in particular) yield very good results with very little grammar and spelling errors. Thus we can present the use of these models as a very promising avenue for auto-regressive text generation for low-resource languages, at least for those that are included in the pre-trained models.

2 Related Work

2.1 MWP Generation

Previous research has addressed the problem of MWP generation using three main techniques: question rewriting, template-based generation and text generation with Neural Networks (NNs).

Question rewriting technique rewrites a human-written question by replacing words with new ones from different contexts (Koncel-Kedziorski et al., 2016). However, the numerical values in all the rewritten questions are the same.

In the template-based techniques, first a question template is either provided by a tutor (Nandhini and Balasundaram, 2011; Polozov et al., 2015; Wang and Su, 2016), or generated from an MWP (Bekele, 2020). Most of these template-based techniques are long and tedious processes, with some requiring language specific tools or resources.

[1] https://huggingface.co/datasets/NLPC-UOM/MWP_Dataset

Zhou and Huang (2019) present a Deep Neural Network model that has two encoders and one decoder, all based on RNNs. The equation encoder takes in an equation template, and the topic encoder takes in a topic (context). The system is trained in a supervised manner, using an MWP dataset. Thus, for training purposes, the equation and the topic of each training MWP has to be extracted. Wang et al. (2021) also take in an equation and context, however MWP generation is done using GPT-2. Additionally, they introduce constraints to satisfy equation and context correctness. Liu et al. (2020) also take in an equation as the input. However, they expect an external knowledge graph to represent the context. Both the knowledge graph and the equation are encoded using a Convolutional Gated Neural Network model. A Variational Auto-Encoder (VAE) is used to generate the MWP from this encoding. Cao et al. (2021) also make use of a VAE to bridge the gap between abstract math tokens and text. In addition to the equation and common sense knowledge graph as input, they take in the question text, as well as a set of words representing a topic.

In contrast to above research, Liyanage and Ranathunga (2019, 2020) train a single RNN encoder in an auto-regressive manner using the MWP text. Liyanage and Ranathunga (2019) impose Mathematical constraints during post processing, while Liyanage and Ranathunga (2020) achieve the same using POS embeddings as input to the model. As for NN-based solutions, only Liyanage and Ranathunga (2019, 2020) considered MWP generation in languages other than English.

2.2 Bench-marking NLG with Pre-trained Models

NLG is an umbrella term used for a set of tasks where the objective is to generate a text as the output. In addition to auto regressive text generation, NLG covers tasks such as text summarization, text simplification, and graph to text generation.

The GEM benchmark (Gehrmann et al., 2021) evaluates BART, T5, mBART and mT5 for 11 different NLG tasks. However, there is no evaluation on an auto regressive text generation task. Moreover, except for one dataset, all the others are focused only on high-resource languages. The GLGE benchmark (Liu et al., 2021), which evaluated BART and MASS pre-trained models also does not have a dataset for auto regressive text generation. Further, evaluation is only done for English.

Several shared tasks have been organized for multilingual NLG tasks such as surface realization (Mille et al., 2020) and RDF triples to text (Ferreira et al., 2020). Submissions to these shared tasks have experimented with various pre-trained models. However, the datasets focus only on high and mid-resource languages. In contrast to the above datasets, Kumar et al. (2022)'s multilingual NLG dataset suit covers many low-resource Indic languages. They use mT5 and IndicBART for evaluation. However, an auto regressive text generation task is not included in this suit. As for auto-regressive text generation evaluation, we are only aware of Chen et al. (2021), who considered mT5 and mBART. However, evaluation was done only on 4 high-resource languages.

3 Methodology

All the models considered in this research are trained using the Transformer architecture (Vaswani et al., 2017), which is an Encoder-Decoder model that contains a set of encoder layers and decoder layers. GPT, BART and T5 are pre-trained with English data. mBART and mT5 are pre-trained with data from multiple languages (50 and 101, respectively). Here, pre-training means, the models have been trained with a self-supervised objective such as 'span corruption' (Xue et al., 2020). All these models have to be fine-tuned for the selected downstream task.

GPT models are decoder based. Here, the encoder-decoder cross attention block is discarded because there is no encoder. Self-attention has been replaced by masked self-attention. We follow the standard training procedure of GPT-2 model in training it for MWP generation. T5, BART, mBART and mT5 are encode-decoder models. They expect a text sequence as the input and output. For auto-regressive text generation, we use the conditional generator option of BART/mBART and T5/mT5, which makes the output of the model conditioned on the preceding input sequence. In both these cases, the models generate the rest of the MWP for a given seed.

4 Experiments

4.1 Dataset

Liyanage and Ranathunga (2020)'s dataset contains two types of MWPs: simple MWPs and algebraic MWPs. The simple MWP dataset contains 2000 questions and the Algebraic MWP dataset contains 2350 questions. This dataset contains questions in English, Tamil and Sinhala, but is not multi-way parallel.

We extended this dataset using the Dolphin18K dataset (Huang et al., 2016) and the allArith dataset (Roy and Roth, 2016) to add more diversity to the dataset. We selected questions that are similar or slightly higher in complexity compared to Liyanage and Ranathunga (2020)'s corpus. Questions that have lengthy descriptions and those corresponding to complex Mathematical equations were omitted. The extended dataset now contains 4210 Algebraic MWPs and 3160 simple MWPs. Simple MWP dataset contains simple arithmetic questions as the example shown in the introduction. These questions contain constraints such as '*first number is always larger than the second one*'. Algebraic MWPs are more logical and require two or more equations to solve.

E.g.: *The sum of two numbers is twenty-three, and the larger number is five more than the smaller number. Find these numbers.*

Corresponding Sinhala and Tamil examples are given in the Figure 1 in Appendix.

Table 1: Statistics of the multi-way parallel dataset

Dataset type	Avg. Num. of words per question	Avg. Num. of characters per question
English Simple (ES)	15	54
English Algebraic (EA)	14	62
Sinhala Simple (SS)	19	61
Sinhala Algebraic (SA)	17	59
Tamil Simple (TS)	13	49
Tamil Algebraic (TA)	16	57

Mathematics tutors translated these questions to Sinhala and Tamil. They were asked to retain the same sentence count and syntactic structure as the English source question, as much as possible. On average, there are two sentences per question, with a maximum of four sentences. Other statistics of the dataset are given in Table 1.

In order to verify the quality of the manual translations, we used the Direct Assessment (DS) method (Bojar et al., 2016). We selected three bilingual speakers (undergraduate students who are proficient in Mathematics) for each language pair (English-Sinhala, English-Tamil). Each evaluator was assigned 200 translated MWPs along with the original English question. They were asked to rate the translated version with respect to adequacy and fluency and give a rating between 1-100, where 0-10: incorrect translation, 11-29: a translation with few correct keywords, but the overall meaning is different from the source, 30-50: a translation with major mistakes, 51-69: a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors, 70-90: a translation that closely preserves the semantics of the source sentence and 91-100: a perfect translation (Bojar et al., 2016). As shown in Table 2, except for the Tamil Algebraic dataset, all the others report a quality level greater than 85.

Table 2: Quality estimation results of the translated dataset

Data set	Rank					
	0-10	11-29	30-50	51-69	70-90	91-100
SS	0%	1.6%	3%	6.3%	22.6%	66%
SA	0%	0%	0.3%	2.6%	12.6%	84.3%
TS	0%	1%	4%	8.3%	27.6%	59%
TA	7%	12%	6.3%	6%	11.3%	57%

Table 3: Language Coverage of pre-trained models

Model		English	Tamil	Sinhala
BART	Storage(GB)	160	-	-
T5	Storage(GB)	700	-	-
mT5	Token(B)	2733	3.4	0.8
	Pages(M)	3,067	3.5	0.5
mBART	Token(B)	55.61	0.595	0.243
	Storage(GiB)	300.8	12.2	3.6

4.2 Model Selection

According to Huggingface[2], GPT2-Medium, T5-base and BART-large variants have approximately 300M model parameters. Therefore these were used for further experiments. For multilingual MWP generation, we selected mT5-base and mBART50-large models, to correspond to their monolingual counterparts. As shown in Table 3, Sinhala and Tamil are largely under-represented in both multilingual models.

4.3 Experiment Setup

Fine-tuning for the selected Huggingface models was set-up with 20 epochs, 4-batch size and 1e-4 learning rate. All the experiments were done on a system that has 15 Intel(R) Core(TM) i9-9900K CPUs and Quadro RTX 6000 GPU with 24GB memory.

[2] https://huggingface.co/transformers/v3.3.1/pretrained_models.html

4.4 Evaluation Metrics

Test BLEU (Papineni et al., 2002) and ROUGE (ROUGE-1 and ROUGE-2) (Lin, 2004) scores were used as the automatic evaluation metrics, as they are still very commonly used (Gehrmann et al., 2021). For all the experiments, we use BLEU-1 for results analysis, with ROUGE results reported in the Appendix. We note that results reported via these two metrics show a correlation.

The generated MWPs should have correct spelling/grammar and satisfy different Mathematical constraints. A Maths tutor should be able to edit a generated MWP in less time compared to writing a question from scratch. We carried out a human evaluation to validate the quality of the generated questions and the time taken by a tutor to correct a generated MWP.

5 Results and Evaluation

5.1 Pre-trained models vs Baseline

Since Liyanage and Ranathunga (2020) have provided the evaluation results for their dataset of English, Tamil and Sinhala, we considered this as our baseline. Our first experiment is to determine whether fine-tuning the pre-trained models is better than the selected RNN baseline.

For this experiment, we used only Liyanage and Ranathunga (2020)'s dataset, and used the same data split (train:validation:test 80:10:10) they have used[3]. Note that for English, results are obtained using the monolingual models.

As mentioned earlier, during training and inference of auto-regressive text generation models, the input to the model is the initial portion of text. This is called a *seed*. In this experiment, we tested our models with a quarter of a question (quarter seed). In contrast, Liyanage and Ranathunga (2020) used the first (50-100) characters. Usually, this attributed to more than half of the question. Note that this means the length of the seed varies from question to question.

Results are shown in Table 4. All our models, even when using just the quarter seed, outperform the baseline by a significant margin, thus highlighting the robustness of the pre-trained models even for low-resource language text generation. Sample questions generated from the models are shown in Table 5. Here, compared to the output of the pre-trained models, the question generated by the baseline is incomplete, not in a question format and has spelling errors.

Table 4: BLEU for the baseline experiments of English, Sinhala and Tamil MWPs.

Dataset type	Model	Seed size	En	Si	Ta
Simple	Baseline	>Half	22.97	24.49	20.74
	GPT-2	Quarter	67.00	-	-
	BART/ mBART	Quarter	80.93	**74.52**	**71.07**
	T5/ mT5	Quarter	**88.42**	68.02	66.45
Algebraic	Baseline	>Half	33.53	-	-
	GPT-2	Quarter	48.93	-	-
	BART/ mBART	Quarter	62.99	**58.13**	**68.21**
	T5/ mT5	Quarter	**72.69**	47.19	55.33

5.2 Effect of Fine-tuning Dataset Size

We conducted comprehensive experiments on our models to analyze how the quality of the results varies with different fine-tuning dataset sizes. We split the dataset for train:validate:test in such a manner that the training set has 80, 40, and 20 percent of the total dataset per MWP category, and conducted three exeriments. Validation and test sets were always kept to be 10% of the total dataset per MWP category. Results are shown in Table 6.

The obvious observation is that the performance of all the models drop when the fine-tuning dataset size drops, which of course is not surprising.

As for English auto-regressive text generation results with monolingual models, both sequence-to-sequence models outperform GPT-2. This is in line with observations for other types of text generation tasks such as graph-to-text generation and question answering (Ribeiro et al., 2021). Further, T5 outperforms BART. We believe this is due to T5 being trained with more data, and this observation confirms with what has been reported for tasks such as machine reading comprehension (Tanaka et al., 2021) and text summarization (Garg et al., 2020). English results with mBART and mT5 lag behind their monolingual counterparts. This is to be expected - the multilingual models do not have English data in the same quantities as their monolingual counterparts. However, this lag is usually around 2 BLEU.

As for multilingual models, mBART outperforms mT5 in all the cases except for the 20% train set scenario of the English Algebraic dataset.

[3] They reported results only using BLEU

Table 5: Sample English MWPs generated using the baseline and the fine-tuned models. Seed size: Quarter of the question

Model	Generated MWPs
Reference	The sum of two numbers is 56, their difference is 22, Find the larger number.
Baseline	the sum of two numbers is 12. their differenct are the two consecutive integers if the sum of the second integers is 10.
Fine-tuned GPT2	The sum of two numbers is 76, the second is 8 more than 3 times first, what are these 2 numbers?
Fine-tuned BART	The sum of two numbers is 60. three times the smaller number minus twice the larger number is 56. Find the larger number.
Fine-tuned T5	The sum of two numbers is 91. the larger number is 1 more than 4 times the smaller number. Find the numbers.

Table 6: Effect of the fine-tuning Dataset Size reported in BLEU (for quarter seed length)

Dataset size	Train Size	Test Size	English					Tamil		Sinhala	
			GPT2	BART	T5	mBART	mT5	mBART	mT5	mBART	mT5
ALG 4210	3370 (80%)	420 (10%)	55.88	60.22	65.32	67.06	62.78	52.68	50.65	45.46	42.44
	1679 (40%)	420 (10%)	54.23	57.76	62.2	60.76	58.86	50.344	49.34	42.58	38.32
	835 (20%)	420 (10%)	51.87	54.93	59.64	53.27	56.34	47.37	42.26	41.03	34.26
SIM 3160	2530 (80%)	316 (10%)	57.65	65.13	67.82	67.74	66.67	65.85	61.67	65.44	61.71
	1264 (40%)	316 (10%)	55.56	57.99	64.43	64.08	62.25	60.24	58.60	60.48	54.08
	632 (20%)	316 (10%)	54.48	55.52	62.09	61.47	57.13	59.5	53.87	56.81	50.92

This is surprising, because as reported in Table 3, mT5 has more Sinhala and Tamil data compared to mBART. Noting that mT5 has more language coverage than mBART, one possible reason for this could be the problem of *curse of multilinguality* - where the cross-lingual transfer in a multilingual model degrades when the language coverage increases in a model (Conneau et al., 2019).

5.3 Effect of Pre-training Dataset Size

An interesting observation is that, although the dataset is multi-way parallel, the result of a model for the same train-test split is not the same across languages. This difference is the highest for the algebraic dataset. Specifically, always English has the highest result, followed by Tamil, and then Sinhala. We attribute this to the amount of language data included in model pre-training (refer Table 3). Moreover, the results gap between Sinhala and English is higher for mT5 compared to mBART. This could be due to the effect of curse of multilinguality that we mentioned earlier - suficient cross-lingual transfer does not happen between Sinhala and English due to mT5's high language coverage.

5.4 Effect of the Context of MWPs

We note that all the models find the algebraic MWP generation more difficult than simple MWP generation. This indicates that text generation capabilities of pre-trained models depend on the context of the text - algebraic MWPs have more Mathematical context than the simple MWPs, which contain more open-domain text that is similar to the text used to pre-train the models.

This may be the reason for the simple MWP dataset to have less language-wise difference in model performance compared to the Algebraic dataset as discussed above - the maximum difference is about 5 BLEU between the best performing English and least performing Sinhala. Given the context of simple MWPs is more similar to the pre-training data, simple MWP generation benefits better from cross-lingual knowledge transfer between related languages.

In order to further evaluate this effect, we carried out an additional experiment - for the 40%-50% train-test split, we trained the models with one dataset, and tested with the other. Results are reported in Table 7. Compared to the results reported in Table 6, we see a substantial drop in the results, when the models are fine-tuned with the other dataset. This highlights the model's inability to generalize to the general problem of MWP generation, if the dataset contains MWPs only representing a specific context.

Table 7: BLEU score results for different domain train and test sizes

Train ID	Train Size	Test ID	Test Size	mBART	mT5
SA	1679 (40%)	SS	1580 (50%)	32.39	29.23
SS	1264 (40%)	SA	2088 (50%)	27.01	17.87
TA	1679 (40%)	TS	1580 (50%)	35.27	33.44
TS	1264 (40%)	TA	2088 (50%)	32.12	27.75

5.5 Zero-shot MWP Generation

Motivated by the results we obtained in Table 6 for small amounts of fine-tuning data, we carried out zero-shot text generation experiments. However, as seen in Table 8, all the models miserably fail on zero-shot text generation. The sample generations shown in Table 9 evidence that the generated sentences are not questions but more like stories. This is because these pre-trained models are not specifically trained on a question-type dataset. However, when fine-tuned with just 100 data samples, the performance increases by a significant margin. This result agrees with the observations of Burnyshev et al. (2021) on few-shot text generation of task-oriented utterances. This provides a ray of hope for low-resource languages - at least for those that are covered by pre-trained multilingual models, even with a very small training dataset, a descent result can be expected. We also note that this zero-shot/few-shot observation in in-line with those reported for other pre-trained models such as mBERT (Lauscher et al., 2020).

5.6 Effect of Seed Length

The next experiment is to determine the impact of seed length. For this, we fixed the train set size to 40% and tested with 50% of the dataset. Experiments are run on mBART, which is shown to outperform mT5. We varied the seed length from 10%-40%. Table 10 reports the results for Sinhala and Tamil. As expected, the quality of the generated text goes up when the seed length increases. However, even 10% of the seed is enough to provide an acceptable result - the lowest is 30 BLEU reported for Sinhala Algebra MPW dataset. The impact of question type and the pre-training data amount of the language can be seen here as well.

Table 8: Zero-shot and few-shot results for Sinhala and Tamil

Test Dataset	Train Size	Test Size	mBART	mT5
ES	0	986	5.96	0.05
EA	0	1175	8.50	0.42
SS	0	986	6.37	0.01
SA	0	1175	7.50	0.03
TS	0	986	4.57	0.02
TA	0	1175	6.54	0.03
ES	100	986	23.24	4.30
EA	100	1175	34.50	3.93
SS	100	986	52.72	5.42
SA	100	1175	18.21	2.36
TS	100	986	48.86	2.87
TA	100	1175	39.95	0.60

Table 9: Sample Zero shot Generation results

Model	Generated MWPs
Reference	The difference between two numbers is 24, Find the numbers if their sum is 88.
GPT2	**The difference between** a "first," and an ordinary, job is that the former often requires significant skills.What's next?Well. . . not much really right nowthough!
BART	The... **The difference between** the two
T5	**The difference between** the two is that the difference between the two is the difference between the

Table 10: Text generation results for different seed sizes

Seed size	SS	TS	SA	TA
10%	48.9	45.48	30.19	36.77
20%	58.25	57.74	39.91	45.82
30%	65.47	65.02	47.38	54.21
40%	71.51	72.39	53.85	62.5

5.7 Human Evaluation

We analysed the questions generated by the different models to identify the types of errors in MWP generation. The identified errors are given in Table 11.

We also wanted to identify the actual utility of

Table 11: Identified errors in the generated MWPs

Error Type	Description	Example
Co-reference	inconsistent co-reference	***Murali*** *had 9 balls in his house and his friend gave him 4. How many balls does* ***Sam*** *have?.* Here, the second sentence has the proper noun Sam, instead of Murali
Unit	A numerical quantity is associated with an inconsistent unit	*Kamal built a house and he used 90 kg cement and* ***40 l sand****. How much more cement than sand did Kamal use?.* Here, sand is given the unit liter (l), insted of kg
Spelling	Spelling mistakes in a word	*What three consecutie odd integers have a sum of -105?* Word 'consecutie' is mispelled.
Grammar	A sentence has grammar mistakes	*The* ***difference*** *of the squares of a number and 6* ***are*** *18. Find the number.* Here, the noun 'difference' is associated with the auxiliary verb 'are'.
Math constraints	The given numerical values do not lead to a meaningful Mathematical equation	*The sum of three consecutive odd integers is 194, what are the integers?* This question cannot be solved without changing the values

the generated questions - whether it is more effective for a tutor to correct a generated question, rather than generating a question from scratch. This experiment was conducted only for Sinhala and English, considering mBART-large and mT5-base models. We gave 20 MWPs (10 simple MWPs and 10 Algebraic MWPs) generated by both mBART and mT5 using 50:40 train:test fine-tuning dataset sizes for quarter input seed to 5 university students[4] who are proficient in English and Sinhala. They were asked to record the time taken to correct each question (refer Table 12 & 13). Then they were given the list of errors we identified in Table 11, and were asked to mark the type of errors they identified. Results of the manual analysis are reported in Tables 14. Note that one generated question may contain more than one type of error.

Table 12: Time taken for a human to correct Simple MWPs (reported in minutes). TTE: Time to Edit 10 generated MWPs, TTG: Time To Generate 10 MWPs

					mBART		mT5	
	TTG		TTE		TTE		TTE	
	SE	SS	SE	SS	SE	SS	SE	SS
T1	18	15	2	2.5	0.5	0.38	0.66	0.66
T2	20	25	2.2	3	0.75	0.45	0.48	0.58
T3	15	17.5	1	1.5	0.55	0.38	0.71	0.51
T4	15	28	2.5	1	0.6	0.83	0.6	0.75
T5	21	26.5	3	2	0.63	0.91	0.45	0.6
Av	**17.8**	**22.4**	**2.14**	**2**	**0.60**	**0.59**	**0.58**	**0.62**

Table 13: Human evaluation results for Algebraic MWPs in minutes AE: Algebraic English, AS: Algebraic Sinhala, (Number of minutes taken to Edit 10 generated MWPs)

	mBART		mT5	
	AE	AS	AE	AS
Tutor 1	2	0.66	1.16	2
Tutor 2	0.73	0.65	0.58	0.73
Tutor 3	0.42	0.75	0.83	0.78
Tutor 4	0.9	0.88	1.26	1.41
Tutor 5	1.25	1.08	0.91	0.95
Average	**1.06**	**0.80**	**0.95**	**1.17**

For English MWPs, mT5 model takes the shortest time to correct. For Sinhala MWPs, mBART model takes the shortest time to correct. Note that all these times are less than what Liyanage and Ranathunga (2020) have reported, who in turn have shown that writing questions from scratch takes considerably more time than text generation from their technique.

Table 14: Percentages of different types of errors found in simple MWPs

Errors%	mBART				mT5			
	SE	AE	SS	AS	SE	AE	SS	AS
Co-reference	4	4	6	4	8	2	6	2
Unit	4	1	1	1	2	1	1	1
Spelling	0	0	4	2	2	0	0	2
Grammar	16	12	16	10	8	10	14	10
math constraint %	12	38	22	30	14	22	24	32

Co-reference, unit, spelling and grammar are usually less than 20% even in the worst performing model. However, errors related to Math constraint violations are relatively high. This implies that the pre-trained models do not have sufficient information to capture constraints specific to a domain, which of course is not surprising.

6 Conclusion

We evaluated several multilingual and monolingual pre-trained models for the task of MWP generation considering four factors - the amount of language-specific pre-trained data, amount of fine-tuning data, length of the seed and type of the MWP. We also presented a multi-way parallel dataset for MWP evaluation, which includes two languages under-represented in these pre-trained models. Our results are very promising - even with a small amount of parallel data and a short seed, all the models are capable of producing acceptable results for all the considered languages. Human evaluation showed that a Mathematics tutor can take benefit of this automated MWP generation, as it saves time compared to writing an MWP from scratch.

In this research, we did not specifically focus on how to satisfy Maths constraints in an MWP. The effect of this was shown in human evaluation - the questions had a noticeable number of issues related to Math constraints. Thus in the future, we plan

[4] Not the same ones who did the translation evaluation

to focus on constraint-based generation of MWP. A starting point would be the work of Wang et al. (2021), who investigated this problem for MWP generation with GPT-2. A major criticism of the pre-trained models is that they support a very small fraction of languages. Thus we want to investigate how the model performance can be improved in the context of languages not included in the model.

7 Ethical Considerations

We have obtained the permission to republish the baseline (Liyanage and Ranathunga, 2020) datasets. In Dolphin18K dataset (Huang et al., 2016) and allArith dataset (Roy and Roth, 2016), they have not mentioned any restrictions on using the data. We cited their papers as requested in their repos. We paid the workers according to the rates defined in our university. We verbally explained the purpose of the dataset and the process they have to follow. Worker information was not collected nor included in the dataset, as this is not relevant to the task. In the fine-tuning process, we only focused on elementary-level MWPs. This dataset is publicly released. It does not have any offensive content, nor specific references to individuals or organizations. Thus the fine-tuning process cannot introduce any additional harmful content to the models. We believe that MWP generation in multiple languages has a long-term positive benefit for school children, and the education sector in general. Thus, the positive impact of this research would outweigh any unforeseen negative impacts it could bring.

8 Acknowledgement

Dataset creation of this project was funded by a Senate Research Committee (SRC) grant of University of Moratuwa (UoM), Sri Lanka. The authors would like to thank the National Language Processing Center (NLPC) of UoM for funding the publication of this paper at INLG.

References

Bed Raj Acharya. 2017. Factors affecting difficulties in learning mathematics by mathematics learners. *International Journal of Elementary Education*, 6(2):8–15.

Andinet Assefa Bekele. 2020. Automatic generation of amharic math word problem and equation. *Journal of Computer and Communications*, 8(8):59–77.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*.

Pavel Burnyshev, Valentin Malykh, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Single example can improve zero-shot data generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 201–211.

Tianyang Cao, Shuang Zeng, Songge Zhao, Mairgup Mansur, and Baobao Chang. 2021. Generating math word problems from equations with topic consistency maintaining and commonsense enforcement. In *International Conference on Artificial Neural Networks*, pages 66–79. Springer.

Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiaze Chen, Hao Zhou, and Lei Li. 2021. Mtg: A benchmarking suite for multilingual text generation. *arXiv preprint arXiv:2108.07140*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional webnlg+ shared task: Overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. 2020. News article summarization with pretrained transformer. In *International Advanced Computing Conference*, pages 203–211. Springer.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016. A theme-rewriting approach for generating algebra word problems. *arXiv preprint arXiv:1610.06210*.

Aman Kumar, Himani Shrotriya, Prachi Sahu, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Amogh Mishra, Mitesh M Khapra, and Pratyush Kumar. 2022. Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages. *arXiv preprint arXiv:2203.05437*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499.

En-Shiun Annie Lee, Sarubi Thillainathan, Shravan Nayak, Surangika Ranathunga, David Ifeoluwa Adelani, Ruisi Su, and Arya McCarthy. 2022. Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? In *Findings of the Association for Computational Linguistics 2022*.

Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983.

Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2021. Glge: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420.

Tianqiao Liu, Qian Fang, Wenbiao Ding, and Zitao Liu. 2020. Mathematical word problem generation from commonsense knowledge graph and equations. *arXiv preprint arXiv:2010.06196*.

Vijini Liyanage and Surangika Ranathunga. 2019. A multi-language platform for generating algebraic mathematical word problems. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 332–337. IEEE.

Vijini Liyanage and Surangika Ranathunga. 2020. Multi-lingual mathematical word problem generation using long short term memory networks with enhanced input features. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4709–4716.

Simon Mille, Anja Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (sr'20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20.

Ahmadreza Mosallanezhad, Kai Shu, and Huan Liu. 2020. Topic-preserving synthetic news generation: An adversarial deep reinforcement learning approach. *arXiv preprint arXiv:2010.16324*.

Kumaresh Nandhini and Sadhu Ramakrishnan Balasundaram. 2011. Math word question generation for training the students with learning difficulties. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, pages 206–211.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Oleksandr Polozov, Eleanor O'Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popović. 2015. Personalized mathematical word problem generation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

Melissa Roemmele. 2016. Writing stories with help from recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Subhro Roy and Dan Roth. 2016. Unit dependency graph and its application to arithmetic word problem solving. *arXiv preprint arXiv:1612.00969*.

Leanne J Rylands and Carmel Coady. 2009. Performance of students with weak mathematics in first-year mathematics and science. *International Journal of Mathematical Education in Science and Technology*, 40(6):741–753.

Bowen Tan, Zichao Yang, Maruan AI-Shedivat, Eric P Xing, and Zhiting Hu. 2020. Progressive generation of long text with pretrained language models. *arXiv preprint arXiv:2006.15720*.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13878–13888.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Patrick W Thompson. 1985. Experience, problem solving, and learning mathematics: Considerations in developing mathematics curricula. *Teaching and learning mathematical problem solving: Multiple research perspectives*, pages 189–243.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang and Zhendong Su. 2016. Dimensionally guided synthesis of mathematical word problems. In *IJCAI*, pages 2661–2668.

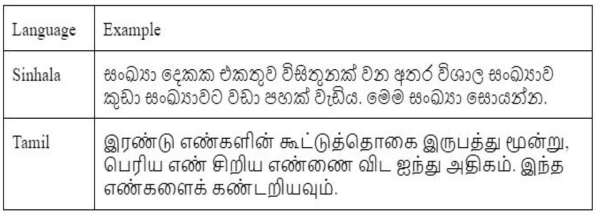
Zichao Wang, Andrew Lan, and Richard Baraniuk. 2021. Math word problem generation with mathematical consistency and problem context constraints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5986–5999.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Qingyu Zhou and Danqing Huang. 2019. Towards generating math word problems from equations and topics. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 494–503.

A Appendix

Figure 1: Sinhala and Tamil Example MWPs



Language	Example
Sinhala	සංඛ්‍යා දෙකක එකතුව විසිතුනක් වන අතර විශාල සංඛ්‍යාව කුඩා සංඛ්‍යාවට වඩා පහක් වැඩිය. මෙම සංඛ්‍යා සොයන්න.
Tamil	இரண்டு எண்களின் கூட்டுத்தொகை இருபத்து மூன்று, பெரிய எண் சிறிய எண்ணை விட ஐந்து அதிகம். இந்த எண்களைக் கண்டறியவும்.

.

Table 15: Zeroshot result ROUGE score for Sinhala and Tamil

Test Dataset	**Train Size**	**Test Size**	**mBART**		**mT5**	
			R-1	R-2	R-1	R-2
ES	0	986	0.467	0.342	0.026	0.005
EA	0	1175	0.439	0.322	0.022	0.003
SS	0	986	0.411	0.275	0.013	0.001
SA	0	1175	0.378	0.248	0.010	0.001
TS	0	986	0.423	0.286	0.007	0.001
TA	0	1175	0.363	0.247	0.005	0.001
ES	100	986	0.241	0.172	0.057	0.024
EA	100	1175	0.352	0.129	0.117	0.022
SS	100	986	0.539	0.362	0.156	0.048
SA	100	1175	0.212	0.074	0.050	0.010
TS	100	986	0.494	0.221	0.076	0.018
TA	100	1175	0.411	0.189	0.031	0.001

Table 16: ROUGE score results for different domain train and test sizes

Train Dataset	**Train Size**	**Test Dataset**	**Test Size**	**mBART**		**mT5**	
				R-1	**R-2**	**R-1**	**R-2**
SA	1679 (40%)	SS	1580 (50%)	0.354	0.246	0.372	0.249
SS	1264 (40%)	SA	2088 (50%)	0.301	0.193	0.271	0.142
TA	1679 (40%)	TS	1580 (50%)	0.384	0.276	0.467	0.324
TS	1264 (40%)	TA	2088 (50%)	0.355	0.253	0.323	0.209

Table 17: Effect of the fine-tuning Dataset Size reported in ROUGE (for quarter seed length)

Dataset size	**Train Size**	**Test Size**	**English**										**Tamil**				**Sinhala**			
			GPT2		**BART**		**T5**		**mBART**		**mT5**		**mBART**		**mT5**		**mBART**		**mT5**	
			R-1	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**	**R-1**	**R-2**
ALG 4210	3370 (80%)	420 (10%)	0.61	0.44	0.61	0.42	0.66	0.50	0.68	0.53	0.65	0.47	0.56	0.40	0.54	0.36	0.49	0.30	0.48	0.28
	1679 (40%)	420 (10%)	0.60	0.42	0.59	0.39	0.64	0.62	0.63	0.46	0.61	0.43	0.54	0.38	0.53	0.36	0.46	0.28	0.44	0.26
	835 (20%)	420 (10%)	0.59	0.51	0.57	0.38	0.62	0.44	0.57	0.38	0.59	0.40	0.51	0.35	0.50	0.34	0.45	0.27	0.42	0.24
SIM 3160	2530 (80%)	316 (10%)	0.64	0.46	0.66	0.51	0.72	0.58	0.72	0.59	0.71	0.57	0.70	0.56	0.66	0.50	0.67	0.52	0.62	0.47
	1264 (40%)	316 (10%)	0.63	0.45	0.61	0.44	0.68	0.68	0.68	0.54	0.66	0.52	0.65	0.49	0.64	0.47	0.63	0.58	0.56	0.53
	632 (20%)	316 (10%)	0.62	0.45	0.59	0.42	0.66	0.52	0.66	0.51	0.62	0.45	0.64	0.48	0.60	0.44	0.59	0.43	0.53	0.36