Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4709–4716
Marseille, 11–16 May 2020

Multi-lingual Mathematical Word Problem Generation using Long Short Term Memory Networks with Enhanced Input Features

Vijini Liyanage, Surangika Ranathunga
University of Moratuwa
Katubedda 10400, Sri Lanka
{vijiniliyanage.12, surangika}@cse.mrt.ac.lk

Abstract

A Mathematical Word Problem (MWP) differs from a general textual representation due to the fact that it is comprised of numerical quantities and units, in addition to text. Therefore, MWP generation should be carefully handled. When it comes to multi-lingual MWP generation, language specific morphological and syntactic features become additional constraints. Standard template-based MWP generation techniques are incapable of identifying these language specific constraints, particularly in morphologically rich yet low resource languages such as Sinhala and Tamil. This paper presents the use of a Long Short Term Memory (LSTM) network that is capable of generating elementary level MWPs, while satisfying the aforementioned constraints. Our approach feeds a combination of character embeddings, word embeddings, and Part of Speech (POS) tag embeddings to the LSTM, in which attention is provided for numerical values and units. We trained our model for three languages, English, Sinhala and Tamil using separate MWP datasets. Irrespective of the language and the type of the MWP, our model could generate accurate single sentenced and multi sentenced problems. Accuracy reported in terms of average BLEU score for English, Sinhala and Tamil languages were 22.97%, 24.49% and 20.74%, respectively.

Keywords: Language Generation, Mathematical Word Problem, LSTM, Embeddings, Low- resource Languages

1. Introduction

A Mathematical Word Problem (MWP) is 'a mathematical exercise, where significant background information on the problem is presented as text rather than in mathematical notation'. (Moyer et al., 1984). Solving an MWP requires knowledge in Mathematics as well as in comprehension.

There is only a limited amount of research done for multi--lingual MWP generation. Existing research either focuses mainly on language-dependent Multiple Choice Questions (MCQs) (Chen et al., 2006), or are template based approaches (Koncel-Kedziorski et al., 2016). Despite this dearth, MWP generation is a challenging task. Since MWPs constitute of a combination of texual facts, Mathematical quantities, units and notations, they tend to contain many constraints. For example consider the MWP:
'Maria made juice and she used 1 litre of water and 0.25 kg of sugar. How much more water than sugar did Maria use? '.

Here the numerical value that represents the amount of water should be of a higher value than that of which represents the amount of sugar. Moreover, relevant units (litre for water and kg for sugar) and appropriate combinations of substances or materials (Eg: water and sugar for juice) should be used. Thus, the generation of MWPs should be done while satisfying these constraints.

In the recent past, deep learning based techniques for Natural Language Generation (NLG) have become popular among many research domains including spoken dialogue systems (Wen et al., 2015), story generation (Roemmele, 2016), lyric generation (Potash et al., 2015), question generation (Zhou et al., 2017), and news generation (Leppanen et al., 2017). NLG models became popular because they facilitate the generation of human readable natural language text, from structured data provided, with less human involvement.

A previous research done by us (Liyanage and Ranathunga, 2019) can be considered as the first work that uses neural NLG techniques for the domain of MWP generation. In that research, we used a character level Long Short-Term Memory network (LSTM) for generating elementary level MWPs. In order to improve the accuracy of the MWPs generated, a post-processing step was introduced, where the generated MWPs were filtered using some hard-coded Part of Speech (POS) rules that check for the satisfaction of constraints based on numerical values and units.

However, this mechanism required us to first identify the constraints available in the MWPs. The POS rules had to be moderated every time a new constraint was found. These rules had to be separately defined for different languages, because the structure of problems differs from one language to another.

Despite the hard-coded nature, that POS based post-processing step showed the importance of POS tags in determining the proper structure of an MWP. Rajpirathap and Ranathunga (2019) have also shown how the type and the context of POS tags could determine the type of an MWP, as MWPs from elementary Mathematics tend to have certain patterns.
In this research, we use POS tag embeddings as input features for the character-level bi-directional LSTM model, which enabled us to fully automate the language-independent MWP generation process. In particular, we used a combination of POS tag and word embeddings, concatenated with character embeddings as input features for the neural model. Then the generation process of the model was further improved through the incorporation of attention on units and numerical values.

We built 8 datasets for the training of the neural model; two single sentenced and multi sentenced simple English datasets, two single sentenced and multi sentenced English Algebraic datasets, two single sentenced and multi sentenced Sinhala datasets and two single sentenced and multi sentenced Tamil datasets. These datasets have been publicly released[1]. The

[1] https://github.com/vijini/MWP_generation.git

new model with the concatenation of embeddings as input features reported improvements in average BLEU-scores of the generated problems by 16.7%, 16.2% and 50.8% for English, Sinhala and Tamil languages respectively, when compared with the baseline model that we used in our previous research (Liyanage and Ranathunga, 2019). The introduction of attention further improved the accuracy of English MWP generation by 13.8% , Sinhala MWP generation by 13.2% and Tamil MWP generation by 27.8% over the baseline model. Our code is publicly available[2].

The rest of the paper is structured as follows. Related work is elaborated in Section 2. Dataset is explained under Section 3. Methodology is provided under Section 4. Evaluation and results are provided under Section 5. Conclusion and Future are provided under Section 6.

2. Related Work

Automatic text generation or Natural Language Generation (NLG) is quite a popular arena in the domain of Natural Language Processing. Automatic text generation has been used for many tasks such as story generation (Roemmele, 2016), factual question generation (M. Heilman, 2011), and lyric generation (Potash et al., 2015).
Early research used rule based techniques such as Conceptual dependency representations (Meehan, 1977) and Knowledge Delivery Systems (Mann and Moore, 1981) for NLG. More recently, neural models such as Recurrent Neural Networks/LSTMs (Graves, 2013), Auto-encoders (Fabius and van Amersfoort, 2014), Reinforcement learning techniques (Guo, 2015), and Generative Adversarial Networks (Goodfellow et al., 2014) have been used.

However, the aforementioned state-of-the-art NLG techniques have not been experimented for MWPs. Since MWPs should be properly examined for the constraints related to Mathematical concepts, numerical values, and units and variable handling, constraint-based language generation is required. Existing approaches (Wang and Su, 2016; Singh et al., 2012; Williams, 2011) for MWP generation are deprived of full automation due to the fact that they are semi or fully template-based. Therefore the MWPs generated by such models follow similar patterns, lacking the creativity and novelty. For an example, Wang and Su (2016) are using previously designed narratives that can be filled up with different numerical values and units that are extracted from a synthesized equation. This makes the generated problems to follow similar patterns or structures. Moreover, generation of MWPs depends on the languages that the templates are written with.

As an alternative, in our earlier research, we presented a neural NLG mechanism to generate elementary level MWPs. There we used a character level LSTM, which could generate multilingual MWPs with an average BLEU-score of more than 80%. In order to make the generated problems 100% accurate, we used a POS based post-processing mechanism, where POS rules were used to filter numerical values, units and adjective-preposition pairs (e.g. 'more than '& 'less than'). It made sure that the generated questions were 100% accurate. However, we had to define the POS rules for each language separately, since the POS tag mappings for MWPs were language specific (This limitation is further elaborated with examples under Section 4.).

[2] https://github.com/vijini/MWP_generation.git

3. Dataset

As shown in Table 1, we used MWPs belonging to three languages, namely, English, Sinhala, and Tamil. All the questions belong to the elementary level, where each question requires simple one or two mathematical operations such as addition, subtraction, multiplication or division. 1,878 questions of the algebraic datasets were extracted from the SigmaDolphin dataset (Shi et al., 2015). Rest of the 472 questions were created manually with the help of some final year undergraduate students of a Computer Science & Engineering department. They have referred Sri Lankan GCE Ordinary Level past papers to find similar questions and have altered them in an appropriate manner. Similarly, Sinhala and Tamil MWP datasets were created.

Language	Question type	No. of questions	single/ multi-sentenced
English	Simple	1350	Single
English	Simple	1350	Multi
English	Algebraic	2350	Single
English	Algebraic	2350	Multi
Sinhala	Simple	1000	Single
Sinhala	Simple	1000	Multi
Tamil	Simple	1000	Single
Tamil	Simple	1000	Multi

Table 1: Stats of the Datasets

These datasets possess different constraints. Some of the constraints identified in the datasets are listed below:

1. Constraints related to the quantities used. Examples include:

- Kamal had 10 balloons and he gave Fred 4 of the balloons, how many balloons does he now have?

Here, the first numerical value should be higher than the second.

2. Constraints related to the units applicable. Examples include:

- Amal made bread and he used 12kg flour and 15l water. How much less flour than water did Amal use?

Here, appropriate units should be used (e.g. kg for flour and l for water).

3. Combination of ingredients/ materials should be chosen appropriately. Examples include:

- Mia built a house and he used 90 kg cement and 40 kg sand. How much more cement than sand did Mia use?

In this example, cement and sand are a couple of materials required for construction of houses.

Consider the following similar example in Sinhala language:

- රෝසි රොටී සෑදූ අතර ඇය පිටි 3kg සහ වතුර 0.5l භාවිතා කළාය. වතුර වලට වඩා වැඩි පිටි කොපමණ ප්‍රමාණයක් රෝසි පාවිච්චි කළේ ද?

  (rōsi roṭī sædū atara æya piṭi 3kg saha vatura 0.5l bhāvitā kaḷāya. vatura valaṭa vaḍā væḍi piṭi kopamaṇa pramāṇayak rōsi pāvicci kaḷē da?)

  Translation- Rosie made roti and she used 3kg flour and 0.5l water. How much flour did Rosie used more than water?

Here, although the nature of the question is the same as of English, the structure is different. Therefore it requires to map the relationship between quantities and units, numerical constraints and the combination of materials used in a language specific manner. For an example in Sinhala questions, the amount of a substance or material together with its units are given after the substance name (Eg: පිටි 3kg). But in the case of English MWPs, the structure is represented as '3kg flour '. Since the structures of problems are language dependant, we have to apply language-specific POS and attention mechanisms on the questions.

4. MWPs should not invalidate mathematical concepts. Examples include:

- The sum of two numbers is 825. Dividing the larger number by the smaller number yields 8 with a remainder of 15. What are the 2 integers?

The numerical values chosen in the above question should be able to produce a couple of simultaneous equations, which once solved will give two integers as the answers.

Furthermore, MWPs possess language specific constraints. For an example, consider the MWP, 'In a car parking area there are 40 cars as blue and red cars. 15 of them are red. How many blue cars are there?'. This problem can be demonstrated in Sinhala and Tamil languages as follows,

In Sinhala language:

- වාහන නැවැත්වීමේ ස්ථානයක නිල් සහ රතු කාර් වලින් කාර් 40 ක් ඇත. ඒවායින් 15 ක් රතු ය. නිල් කාර් කීයක් තිබේද?

- Transliteration: vāhana nævætvīmē sthānayaka nil saha ratu kār valin kār 40 k æta. ēvāyin 15 k ratu ya. nil kār kīyak tibēda?

In Tamil language:

- **கார் பார்க்கிங் பகுதியில் நீல மற்றும் சிவப்பு கார்களில் 40 கார்கள் உள்ளன. அவற்றில் 15 சிவப்பு. எத்தனை நீல நிற கார்கள் உள்ளன?**

- Transliteration: Kār pārkkin pakutiyil nīla marrum civappu kārkaḷil 40 kārkaḷ uḷḷana. Avarril 15 civappu. Ettanai nīla nira kārkaḷ uḷḷana?

Therefore we can easily visualize that the language structures are different. For an example, '40 cars' in English is represented as ' කාර් 40 ක් ', and '**40 கார்கள்**' in Sinhala and Tamil, respectively. Therefore there is a requirement of a language independent model for automatic MWP generation.

4. Methodology

The architecture of our previous solution (Liyanage and Ranathunga, 2019) is represented in the the top part of Figure 1, which used a character level Long Short Term Memory Network with a batch size of 128, 15 epochs and softmax activation. We trained the model by splitting the datasets within Train : validate : test with a ratio of 80 : 10 : 10. Initially, we input a seed text of 50 - 100 characters (e.g. Winston made pudding and he used 9 kg white flour and), which is randomly chosen from the patterns identified for each dataset. The model is capable of generating the rest of the characters until the full MWP is created.
In this system, we had to identify all the specific constraints and manually define the POS rules required to resolve the identified constraints. Every time a new constraint was found, we had to change the rules and check for the results. The constraints found in datasets are language dependent (refer Section 3. for the identified constraints) as well. Consider the following example for POS tagged sentences in different languages,

- If| IN Saran| NNP buys| VBD 16kg| CD of| IN rice| NN and| CC gives| VBZ 6kg| CD of| IN it| PRP to| TO his| PRP brother| NN, how| WRB much| JJ rice| NN does| VBZ he| PRP have| VB ?

- සරන්| NNP සහල්| NNC 16kg| NUM ක්| RP මිලදී| VNF ගෙන| VNF ,| PUNC එයින්| PRP 6kg| NUM ක්| RP මල්ලීට| NNC දුන්නේ| VP නම්| POST ,| PUNC ඔහු| PRP සතුව| VNF ඉතිරි| JJ සහල්| NNC කොපමණද| VP ?| PUNC

  (Saran sahal 16kg k miladī gena, eyin 6kg k mallīṭa dunnē nam, ohu satuva itiri sahal kopamaṇada?)

- **சரன்**| NN 16kg| QC **அரிசி**| NN **வேண்டி**| VM **அதில்**| PRP 6kg| QC **தம்பிக்கு**| PRP **கொடுத்தால்**| VM ,| SYM **அவனிடம்**| PRP **மீதமுள்ள**| JJ **அரிசி**| NN **எவ்வளவு**| RB ?| SYM

  (Caran 16kg arici vēṇṭi atil 6kg tampikku koṭuttāl, avaniṭam mītamuḷḷa arici evvaḷavu?)

For an example, the POS tag sequences representing the material type, its quantity and unit combination are VBD+CD+IN+NN for English MWP, NNC+NUM+RP for Sinhala MWP and QC+NN for Tamil MWP. Therefore it can be seen that even the same MWP translated in different languages have different structures, thereby making the POS tag mappings language specific. Therefore, it is required to separately define post processing POS tag algorithms in a language specific manner.

In order to eliminate the aforementioned limitations in our previous rule based POS tag mechanism, in this research we introduce an end-to-end neural model for MWP generation. We removed the rule-based post processing mechanism and used POS as input features for the model. Further our approach added attention and word embeddings concatenated with character embeddings to improve the model. As depicted in the bottom part of the Figure 1, our novel approach uses a Convolutional Neural Network based LSTM (CNN-LSTM) (Kim et al., 2016) to form character embeddings for each word. Characters of each word are fed as inputs to a 1D convolutional

layer, and its outputs are added to the LSTM model by wrapping the entire sequence of CNN layers in a time distributed layer. Dropout regularization was used on this CNN-LSTM to prevent over-fitting. The character embeddings formed as output of the CNN-LSTM are concatenated with the other embeddings to form the input to the main LSTM model. The latter LSTM is the model that is responsible for the generation of MWPs. It is designed with a batch size of 128, 10 epochs and softmax activation.

POS tags have been used as input features in some research done for Neural Machine Translation (Sennrich and Haddow, 2016), text - based question generation (Zhou et al., 2017), and answer generation for MWPs (Rajpirathap and Ranathunga, 2019). In particular, Rajpirathap and Ranathunga (2019) used a POS tag based feature extraction mechanism to identify whether the first numerical value is greater than the second value. This further highlights that POS is a suitable source of information to resolve constraints related to numerical quantities.

Initially, POS tag embeddings concatenated with word embeddings were used as inputs for the model. Here we used the POS tag mechanism defined using lexical categorization (Loper and Bird, 2002) (which is used in Natural Language Toolkit (nltk)) to generate POS tags for English MWP datasets, and word embeddings were created using FastText. After that, a combination of POS tag embeddings, word embeddings and character embeddings (which were created using a CNN) were used as the summation of input features (I) as shown in the equation 1:

$$I = \sum_{j=1}^{M} \| W_j P_k (\sum_{i=1}^{N} \| C_i) \quad (1)$$

where $\|$ is the vector concatenation. W_j and P_k are the word embeddings and POS tag embeddings of each word, respectively. C_i is the character embedding of each character in a particular word. i, j and k represent the number of characters in each word, the number of words in each sentence, and the number of POS tag embeddings defined for the dataset, respectively.

Application of attention for neural text generation has been popular (Xie, 2017), since the attention mechanism is capable of allowing the decoding function to focus on specific areas in the input, depending on the decoding requirement. Vaswani et al. (2017) have stated attention as an integral part of sequence modeling, because attention facilitates modeling of dependencies irrespective of the distance between input or output sequences. Therefore in our research, we incorporated attention to improve our model by enabling the attention mechanism on numerical values and units of MWPs. For example, consider the MWP, 'Dina made cookies and she used 0.625kg flour and 1.25kg sugar. How much less flour than sugar did Dina use?'. Here, our approach applies the attention on the two numerical values (0.625 & 1.25) and their associated units (kg).

We extended our research for Sinhala and Tamil languages as well. Since Sinhala and Tamil are two morphologically rich languages, application of POS tags should be handled carefully. The POS tag set and the POS tagger introduced by Fernando et al. (2016) and (Fernando and Ranathunga, 2018) were used to tag Sinhala language datasets, while the POS tag set used by Thayaparan et al. (2018) was used to tag the Tamil datasets.

Further we applied Temperature tuning in order to vary the creativity and novelty of the generated questions. Softmax temperature (Buscema, 1998) is a hyper-parameter that is used in neural models to control the entropy of the probability distribution. If the temperature parameter is set to a higher value, the randomness of the predictions increases, making the outputs differ from the input dataset. Therefore the creativity of the generated MWPs will be high.

5. Results and Evaluation

We evaluated the results of our research in terms of both the BLEU-score (Test and Self), and human evaluation. Test BLEU-score measures the accuracy of the MWPs generated by comparing with the dataset, while the Self BLEU-score measures the novelty and the creativity of the MWPs generated by identifying the similarity of different questions formed. A higher Test BLEU represents a higher accuracy, and a lower Self BLEU represents a higher creativity. We measured Test and Self BLEU-scores by adjusting the temperature parameter of the model for a range of values and the trade off temperature (Caccia et al., 2018), and the parameter value that gave the highest Test BLEU and the lowest Self BLEU was chosen.

Figures 2, 3, 4 and 5 show the graphs constructed depicting Negative Test BLEU versus Self BLEU scores with respect to simple English, English algebraic, simple Sinhala and simple Tamil datasets, respectively. The temperature that provides the highest Test BLEU and the lowest Self BLEU, indicating the highest accuracy and highest creativity, respectively is chosen as the trade-off temperature. Trade-off temperature parameter values for simple English, English algebraic, simple Sinhala and simple Tamil were 1.2, 1.2, 1.5 and 1.0 respectively. The trade off temperature was chosen by considering the summation of Test BLEU and negative Self BLEU scores that ensure that the accuracy and creativity are maximized (recall that high Test BLEU stands for high accuracy, while low Self BLEU stands for high creativity).

The BLEU-score results after each experiment are given in tables 2, 3, 4 and 5 regarding simple English, English algebraic, Sinhala and Tamil MWP generations, respectively. We used the optimized character-level LSTM reported in our previous work Liyanage and Ranathunga (2019) as the baseline.

Once the concatenation of word, POS and character embeddings together with attention were applied on the neural model, there were improvements by 33% (from Avg-BLEU of 17.30% to 22.97%), 38% (from Avg-BLEU of 24.22% to 33.53%), 32% (from Avg-BLEU of 18.62% to 24.49%) and 93% (from Avg-BLEU of 10.76% to 20.74%) for simple English, English algebraic, Sinhala and Tamil MWP generations, respectively. It can be argued that the morphological richness of Tamil language has paved way for the incorporation of embeddings (POS + character + word) and attention to increase the accuracy of the generated Tamil MWPs with a higher proportion, when compared with the other languages. Since the POS tag set for Sinhala Language was based on a previous

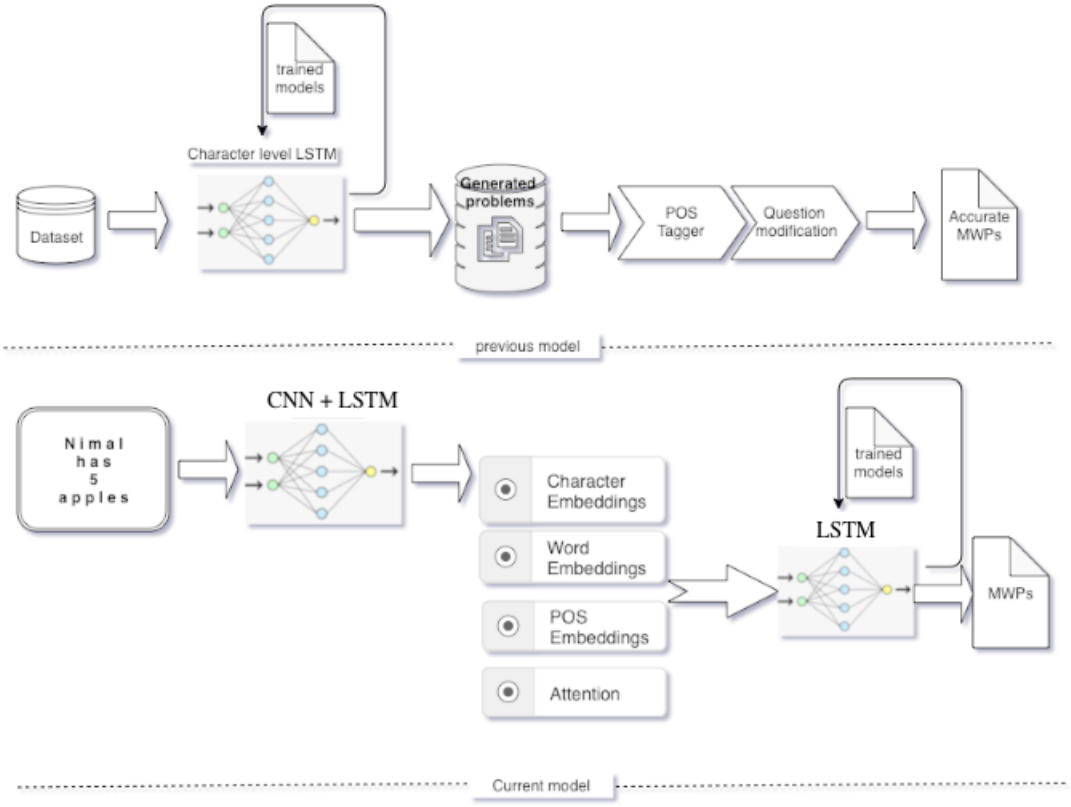

Figure 1: Architecture diagram of two systems.

dataset that was used for a different domain in (Fernando et al., 2016), the improvement of results were not higher as for the generation of English MWPs.

We also used a human evaluation mechanism to validate the outputs generated by our model in terms of adequacy and fluency. We accompanied a group of five tutors to evaluate the generated MWPs in all three languages. Tutors were final year undergraduates of the Department of Computer Science & Engineering, University of Moratuwa, who were conducting classes for Ordinary Level and Advanced Level students in Sri Lanka. Each tutor was asked to correct 10 generated MWPs from each language, if they contain any errors. They were also asked to manually generate 10 fresh problems from each language. They had to keep track of time spans required for each task. These time spans were compared to check the effectiveness of our solution. The results collected with respect to human evaluation are depicted in Table 6. Although Sinhala and Tamil are the mother tongue of the tutors who produced Sinhala and Tamil datasets, the time consumed for the creation of Sinhala and Tamil questions was higher than that for English. This might be due to the fact that it is hard to type in Sinhala and Tamil languages as the tutors are not used to it, when compared with typing in English.

However, the baseline model gave sub-optimal results. For example, with respect to the generated MWPs shown in Table 7, the questions generated with the baseline model got two issues,

1. kg is not a suitable unit to represent the amount of water
2. Since the quantity of cement is smaller than that of water,

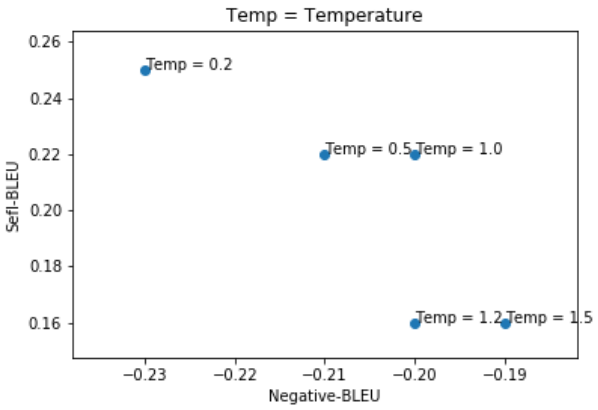

Figure 2: Negative Test-BLEU VS Self-BLEU graph for simple MWPs in English.

Model	BLEU 2	BLEU 3	BLEU 4	BLEU 5	BLEU Avg
Baseline	27.04	21.62	13.21	7.33	17.30
WP	26.98	25.02	12.91	5.87	17.70
WPC	29.37	**28.22**	13.76	9.41	20.19
WPC A	**32.93**	28.01	**16.23**	**14.71**	**22.97**

Table 2: BLEU Scores Produced By Different Models regarding the formation of simple English MWPs. WP: Word + POS embeddings, WPC: Word + POS + Character embeddings, A: Attention

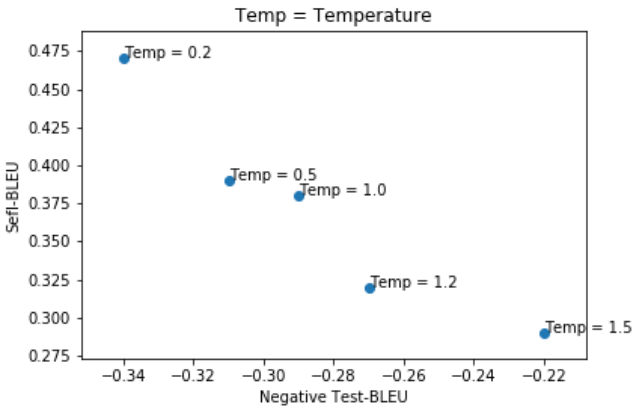

Figure 3: Negative Test-BLEU VS Self-BLEU graph for complex MWPs in English.

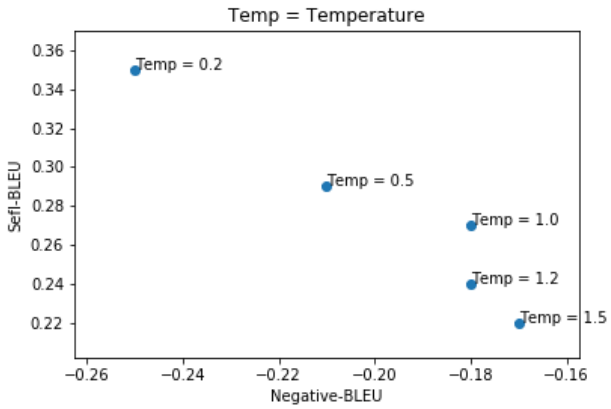

Figure 4: Negative Test-BLEU VS Self-BLEU graph for Sinhala MWPs.

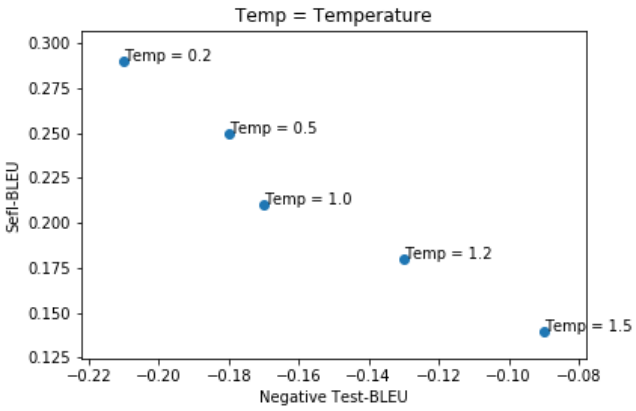

Figure 5: Negative Test-BLEU VS Self-BLEU graph for Tamil MWPs.

it is wrong to ask **more cement than water**.

After using the concatenation of embeddings as input features, the problem with the units was resolved (unit representing amount of water changed from **kg** to **l**).
Finally with the introduction of attention, the issue with the numerical constraint was also resolved by generating a higher numerical value as the quantity of cement than that of for water.

Model	BLEU 2	BLEU 3	BLEU 4	BLEU 5	BLEU Avg
Baseline	37.04	23.89	18.75	17.20	24.22
WP	43.23	25.31	9.72	9.33	21.90
WPC	45.43	31.93	26.23	19.11	30.68
WPC A	**47.84**	**37.03**	**29.42**	**19.83**	**33.53**

Table 3: BLEU Scores Produced By Different Models regarding the formation of Complex English MWPs

Model	BLEU 2	BLEU 3	BLEU 4	BLEU 5	BLEU Avg
Baseline	29.83	19.20	16.72	8.74	18.62
WP	35.73	23.42	16.88	11.73	21.94
WPC	35.23	23.56	17.72	10.02	21.63
WPC A	**39.21**	**25.51**	**21.30**	**11.95**	**24.49**

Table 4: BLEU Scores Produced By Different Models regarding the formation of simple Sinhala MWPs

Model	BLEU 2	BLEU 3	BLEU 4	BLEU 5	BLEU Avg
Baseline	22.91	12.13	7.97	0.02	10.76
WP	25.32	17.48	13.12	5.31	15.31
WPC	24.12	18.59	17.28	4.93	16.23
WPC A	**29.15**	**22.43**	**18.25**	**13.12**	**20.74**

Table 5: BLEU Scores Produced By Different Models regarding the formation of simple Tamil MWPs

6. Conclusion & Future work

In this paper, we presented a multi-lingual elementary level MWP generation model. The model demonstrated that a concatenation of embeddings such as POS, word and character, and the attention mechanism used with a bi-LSTM network is able to satisfy constraints in MWPS to a great extent. Although the structure of elementary level MWPs changes across the used languages, the model was successfully able to identify these language-specific structures during the generation process.

Our research can be considered as the first attempt to automatically generate MWPs in an end-to-end manner, while satisfying the numerical constraints specific to MWPs, as well as the linguistic constraints that are language specific. We have made the MWP datasets and the models we used for MWP generations publicly available.

We used existing POS tagged corpora for Sinhala and Tamil languages that were defined focusing on general datasets. This made some of the POS tag mappings on Sinhala and Tamil MWPs to be irrelevant. Therefore we hope to define POS tagged corpora that are specific to Sinhala and Tamil MWP datasets in our future work.

Currently our approach applies attention only on the numerical values and units. But the accuracy of the generated MWPs depends on the Mathematical constraint satisfaction

	TTG 10 SE MWPs	TTE 10 SE MWPs	TTG 10 CE MWPs	TTE 10 CE MWPs	TTG 10 SS MWPs	TTE 10 SS MWPs	TTG 10 ST MWPs	TTE 10 ST MWPs
Tutor 1	18	2	23	2	15	2.5	20	3.5
Tutor 2	20	2.2	27	3	25	3	19	4
Tutor 3	15	1	28.5	3.5	17.5	1.5	25	3
Tutor 4	15	2.5	22	2.4	28	1	23	2.5
Tutor 5	21	3	23	3.1	26.5	2	30	2.5
Average	17.8	**2.14**	24.7	**2.8**	22.4	**2**	23.4	**3.1**

Table 6: Human evaluation results in terms of TTG (Time To Generate) 10 fresh MWPs VS TTE (Time To Edit) 10 MWPs that are generated by our model
SE: Simple English, CE: Complex English, SS: Simple Sinhala, ST: Simple Tamil

Model	A sample MWP generated
Baseline	Vimal built house and he used 2kg cement and **6kg water**, how much **more cement than water** did vimal use?
LSTM with embeddings	Vimal built house and he used 2kg cement and **6l** water, how much **more cement than water** did vimal use?
With Attention	Vimal built house and he used **5kg** cement and **3l** water, how much more cement than water did vimal use?

Table 7: Sample English questions produced after each mechanism employed

as well. Those constraints are supported by many other POS tag classes such as adverbs and adjectives as well. Therefore we hope to provide attention on a POS tag class level and determine the classes that can make comparatively an impact on the accuracy of the generated MWPs.

At this level, our research was targeted on elementary level MWPs as well as Algebraic MWPs. We hope to extend our research with other categories of MWPs as well. As an initial step, we hope to build multi-lingual MWP datasets for other types of Mathematical problems.

We hope to deliver an end to end system for MWP generation, through which solutions provided by students are also provided for generated MWPs. Some research (Rajpirathap and Ranathunga, 2019) provided solutions for answer generation for MWPs. Therefore we hope to incorporate those models to derive answers for the generated MWPs.

7. Acknowledgements

This research was funded by a Senate Research Committee (SRC) Grant of University of Moratuwa, Sri Lanka. The first author acknowledges the support received from the LK Domain Registry in publishing this paper.

8. Bibliographical References

Buscema, M. (1998). Back propagation neural networks. *Substance use & misuse*, 33(2):233–270.

Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. (2018). Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Chen, C., Liou, H., and Chang, J. (2006). Fast: an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 1–4. Association for Computational Linguistics.

Fabius, O. and van Amersfoort, J. R. (2014). Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*.

Fernando, S. and Ranathunga, S. (2018). Evaluation of different classifiers for sinhala pos tagging. In *2018 Moratuwa Engineering Research Conference (MERCon)*, pages 96–101. IEEE.

Fernando, S., Ranathunga, S., Jayasena, S., and Dias, G. (2016). Comprehensive part-of-speech tag set and svm based pos tagger for sinhala. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 173–182.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Guo, H. (2015). Generating text with deep reinforcement learning. *arXiv preprint arXiv:1510.09202*.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Koncel-Kedziorski, R., Konstas, I., Zettlemoyer, L., and Hajishirzi, H. (2016). A theme–rewriting approach for generating algebra word problems. *arXiv preprint arXiv:1610.06210*.

Leppanen, L., Munezero, M., Granroth-Wilding, M., and Toivonen, H. (2017). Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.

Liyanage, V. and Ranathunga, S. (2019). A multi-language platform for generating algebraic mathematical word problems. *arXiv preprint arXiv:1912.01110*.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

M. Heilman, M. (2011). Automatic factual question generation from text. *Language Technologies Institute School of Computer Science Carnegie Mellon University*.

Mann, W. C. and Moore, J. A. (1981). Computer generation

of multiparagraph english text. *Computational Linguistics*, 7(1):17–29.

Meehan, J. R. (1977). Tale-spin, an interactive program that writes stories. In *Ijcai*, volume 77, pages 91–98.

Moyer, J., Sowder, L., Threadgill-Sowder, J., and Moyer, M. (1984). Story problem formats: Drawn versus verbal versus telegraphic. *Journal for Research in Mathematics Education*, pages 342–351.

Potash, P., Romanov, A., and Rumshisky, A. (2015). Ghostwriter: Using an lstm for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924.

Rajpirathap, S. and Ranathunga, S. (2019). Model answer generation for word-type questions in elementary mathematics. In *International Conference on Applications of Natural Language to Information Systems*, pages 17–28. Springer.

Roemmele, M. (2016). Writing stories with help from recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. *arXiv preprint arXiv:1606.02892*.

Shi, S., Wang, Y., Lin, C., Liu, X., and Rui, Y. (2015). 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.

Singh, R., Gulwani, S., and Rajamani, S. (2012). Automatically generating algebra problems. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Thayaparan, M., Ranathunga, S., and Thayasivam, U. (2018). Graph based semi-supervised learning approach for tamil pos tagging. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, K. and Su, Z. (2016). Dimensionally guided synthesis of mathematical word problems. In *IJCAI*, pages 2661–2668.

Wen, T., Gasic, M., Mrksic, N., Su, P., Vandyke, D., and Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Williams, S. (2011). Generating mathematical word problems. In *2011 AAAI Fall symposium series*.

Xie, Z. (2017). Neural text generation: A practical guide. *arXiv preprint arXiv:1711.09534*.

Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., and Zhou, M. (2017). Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.