Arizona State University
CSE 576 - Natural Language Processing (Fall 2020)
Submission of Phase 1 - Dataset Creation

ASU ID: 1217135725
Shivam Raval

September 24, 2020

- **Introduction:**

The main aim of this phase of the project is to understand the existing available datasets for the given problem and come up with some techniques to generate synthesized dataset as well as explain the model input, ouput along with the metrics to evaluate the model.

1. **Existing Datasets**

The existing datasets that are available for numerical reasoning question answering tasks are mentioned below. Among them I have explored and anlayzed AQuA-RAT, MathQA and DROP dataset in detail.

- AI2 [1] - Single and multistep arithmetic word problems.
- IL [2] - Single-step word problem which requires some common sense
- MAWPS [3] - Testbed for arithmetic word problems with one unknown variable in the question.
- AQUA-RAT [4] - Algebraic question answering dataset with rationale given which explains the method to reach to answer using natural language.
- MAthQA [5] - This dataset is similar to AQUA-RAT but the additional change is that they provide rationale in terms of operations needed to perform to reach to answer (Ex: subtraction(#,addition(no,n1))) instead of a natural language solution like AQUA-RAT.
- DROP [6] - This is different type of dataset where a long paragraph is given and several questions are derived from the paragraph which can be answered using the numerical information given in the paragraph.

2. **Word Math Problem Task**

Understanding a paragraph and inferring or answering questions from it is a challenging task for machine intelligence. Here, in this project we focus on a specific type of question answering problems which requires numerical reasoning to answer the question posed from the paragraph or problem. Due to advances in Natural Language Processing many models have been proposed which are able to answer questions from normal task but answering from compositional paragraphs that require multiple steps of reasoning against text still remains a challenging task especially when they involve discrete and symbolic operations as posed in the word math problems. The main goal of this project is to develop a deep learning model which is able to answer questions with numerical reasoning. The input and output of the model is described

below along with the metrics to evaluate the performance of the implemented model.

Input: A question containing numerical information used to calculate the answer. (Another tpye of input can be a masked number which is computed by the other information given in the paragraph)
Output: A numerical value which is the answer.
Annotations: Annotations can be the explanation to reach to the answer or specific format of operations performed on the input to reach to the answer. It can also be the general information which can be used to calculate the answer. (The examples are shown in the next section).
Evaluation Metric: The evaluation metric to judge the model is the Exact Match (EM) and F1 score which is calculated from precision and recall.

3. **Example Data Samples**

This sections gives some examples written by me which represents a dummy structure of the dataset.

Example 1:
Input/Question: What is the total mass of Uranium Oxide (U_30_8)?
Knowledge Required: Mass or Uranium is 238 and Oxgyen is 16.
Procedure to solution (Rationale): Three amount of uranium (3x238) and eight amount of Oxgyen (8x16)
Options: (A) 714 (B) 128 (C) 842 (D) 900
Output: (C) 842

Example 2:
Input/Question: Player A has n+1 coins, while B has n coins. Both players throw all of their coins simultaneously and observe the number of heads. If all coins are fair, then what is the probability that A obtains more heads than B?
Knowledge Required: Probability should be between 0 and 1 inclusive.
Procedure to solution (Rationale): The probability that A produces more heads than B is independent of the number of coins
Options: (A) 0.5 (B) 0.7 (C) 0.1 (D) 1
Output: (A) 0.5

Example 3:
Input/Question: In a group of 6 boys and 4 girls, four children are to be selected. In how many different ways can they be selected such that at least one boy should be there?
Procedure to solution (Rationale): 4 combinations are possible when at least one boy is to be selected everytime.
Options: (A) 220 (B) 209 (C) 128 (D) 300
Output: (B) 209

Example 4:
Input/Question: A satellite is sent to mars from earth and it travels from mars to moon. What is the total distance in miles travelled by the satellite?
Knowledge Required: The distance from earth to mars (40 million miles) and earth to moon (0.3 million miles).
Procedure to solution (Rationale): Sum the distance from earth to moon plus twice the distance between earth and mars

Options: (A) 40.3 million (B) 80 million (C) 100 million (D) 80.3 million
Output: (D) 80.3 million

Example 5:
Input/Question: There are _[MASK]_ seats in 8 rows and the total number of seats are 80.
Procedure to solution (Rationale): The total number of seats (80) are seats per row(x) multiplied by total rows (8).
Options: (A) 12 (B) 5 (C) 10 (D) 9
Output: (C) 10

4. Synthesized Dataset Generation

This section proposes several methods of generating synthesized dataset. The below sub-sections explain those methods in detail along with the flow diagrams and examples.

1. Generating new Samples using LeakyGAN:

Generative adversarial networks has shown significant progress in computer vision to generate new samples from the training data which are different and can be used for training. LeakyGAN [7] have proposed generative adversarial network to generate text data when some information is give to the network.

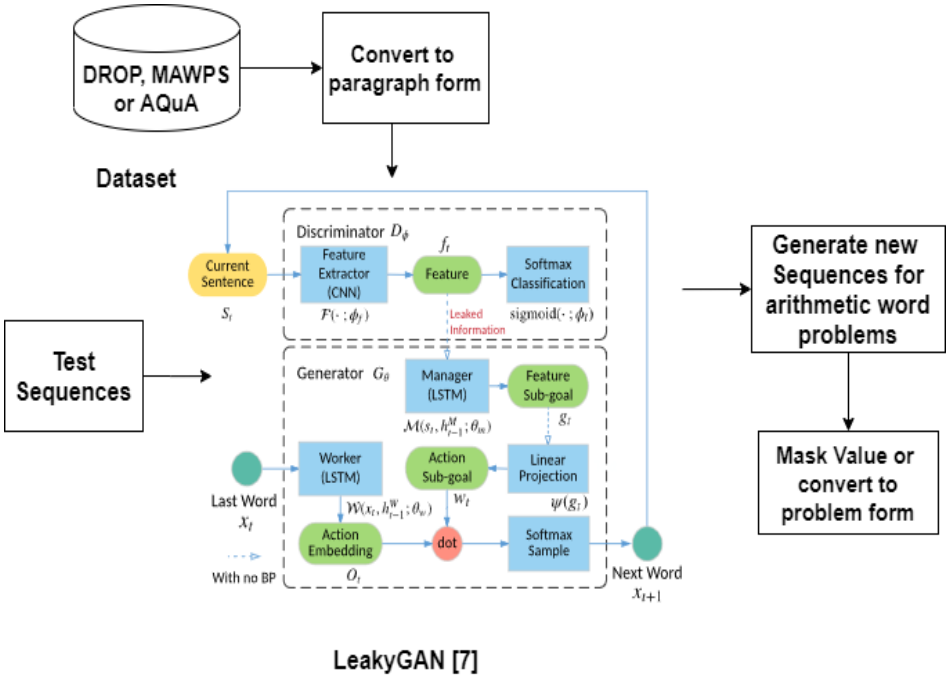

Figure 1: Generating new Samples using LeakYGAN

Fig.1 shows the flow diagram of how the approach would look, first any dataset like MAWPS [3], DROP [6] or AQuA [4] can be converted to a paragraph form by combining the question and answer. After that, the dataset can be passed to the LeakyGAN which basically learns the features from the dataset. In the next step, after the LeakyGAN is trained, test sequences can be passed which is partially empty and LeakyGAN would generate the remaining part of the sentence. The new samples generated by the LeakyGAN would be totally new samples that are derived from the datasets. In the new samples a random number can be masked which can be treated as the output and the model should compute that value. This is just a concept idea that I think is possible but LeakyGAN performance has never been tested on the data

with numerical values. The paper has shown the performance of the model on Oracle and COCO captions dataset and it gives state of the art performance on that datasets so I think it is possible it might give good performance for numerical language dataset. There have been various methods proposed [8], [9] and [10] specifically for math word problems generation by transforming the algebric equation into words given different topics.

2. Using Text Spinner to generate new samples:

New samples can be generated utilizing the datasets mentioned in the first section using Text Spinner. Text Spinner basically changes the words orientation and original words are changed with their synonyms as well as it changes the grammar of the sentence. Text Spinner does not change the meaning of the sentence and the resultant sentence after spinning would have the same core idea as presented in the original sample. This is useful because we can generate whole new samples which the model can interpret in different ways. The procedure to reach to the answer and the output would remain the same since the spinned sentence would raise the same question but in a different manner. There are many text spinner api's available which can be used to generate new samples. A point should be noted that sometimes the sentence generated by text spinning does not have proper grammar so after generating new sample a grammar check should be applied to ensure that the generated sample is grammatically correct. The flow diagram for this method is shown in Fig. 2 which also includes a step to reduce the lexical overlap between two problems. The examples of applying text spinning and new samples generated is shown below:

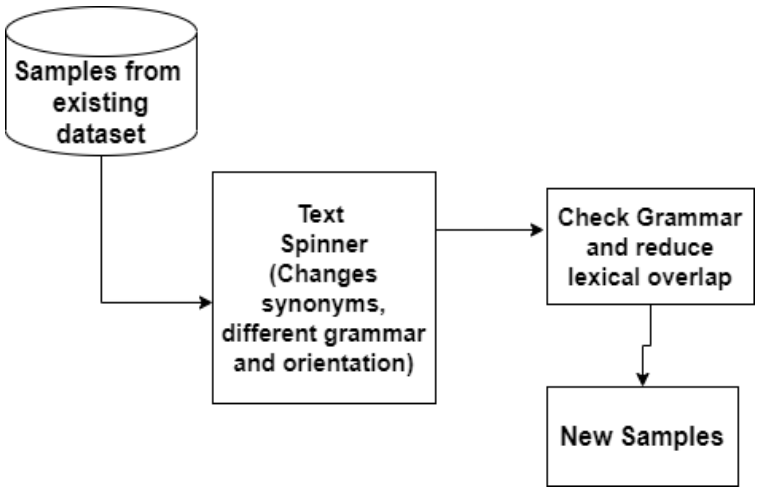

Figure 2: Text Spinning

Examples:

Original Sample: Two friends plan to walk along a 43km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Qś, how many kilometers will Friend P have walked when they pass each other?
Text Spinned Sample: Two companions intend to stroll along a 43km trail, beginning at far edges of the path simultaneously. On the off chance that Friend P's rate is 15% quicker than Friend Qś, what number of kilometers will Friend P have strolled when they pass one another?

Original Sample: The price of a product is reduced by 30% . By what percentage should it be increased to make it 100%

Text Spinned Sample: The cost of an item is diminished by 30% . By what rate should it be expanded to make it 100%

3. Masking and changing the variable to compute:

Another way the Synthesized data can be generated is using the existing dataset and changing the variable to guess. For example, given a question and answer pair, we can incorporate the answer and change it to a paragraph form, after the paragraph is formed we can mask any other random number present in the question so that the computation and reasoning required for calculating the mask number would be different than the original question. A example can be shown below:

Original Sample (From AQuA):
Input: The original price of an item is discounted 22%. A customer buys the item at this discounted price using a $20-off coupon. There is no tax on the item, and this was the only item the customer bought. If the customer paid $1.90 more than half the original price of the item, what was the original price of the item?" **Output:** $78.20

Step 1: Transform to paragraph by answering the question - The original price of an item is discounted 22%. A customer buys the item at this discounted price using a $20-off coupon. There is no tax on the item, and this was the only item the customer bought. The customer paid $1.90 more than half the original price of the item and original price of the item is $78.20.

Step 2: Mask a random number other than the answer - The original price of an item is discounted _[MASK]_ %. A customer buys the item at this discounted price using a $20-off coupon. There is no tax on the item, and this was the only item the customer bought. The customer paid $1.90 more than half the original price of the item and original price of the item is $78.20.
Output: 22 (Incorporate the masked number as a answer to calculate)

One drawback of this kind of method is that for datasets like AQUA-RAT, the rationales defined in natural language would change since the computation that is happening to calculate the masked number would be different. A different kind of annotation can be generated using the method described in MathQA paper [5] which can generate the series of operation needed to compute the masked number.

Apart from this three methods, a new dataset can also be crawled from different online websites but dataset like MAWPS, IL and AI2 are built that way so it would be almost similar to them. Additonally, while looking at different datasets I found out that there are less problems of the type permutations and combinations, so I think a complex dataset can be built which contains easy and hard level problems of permutations and combinations type which requires the model to learn intermediary tasks like probability and other linear operations to compute the answer.

References

[1] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman, "Learning to solve arithmetic word problems with verb categorization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 523–533.

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

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

[4] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program induction by rationale generation: Learning to solve and explain algebraic word problems," *arXiv preprint arXiv:1705.04146*, 2017.

[5] A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "Mathqa: Towards interpretable math word problem solving with operation-based formalisms," *arXiv preprint arXiv:1905.13319*, 2019.

[6] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs," *arXiv preprint arXiv:1903.00161*, 2019.

[7] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," *arXiv preprint arXiv:1709.08624*, 2017.

[8] O. Polozov, E. O'Rourke, A. M. Smith, L. Zettlemoyer, S. Gulwani, and Z. Popović, "Personalized mathematical word problem generation," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

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

[10] Q. Zhou and D. Huang, "Towards generating math word problems from equations and topics," in *Proceedings of the 12th International Conference on Natural Language Generation*, 2019, pp. 494–503.