NumEval: Numeral-Aware Language Understanding and Generation

11/26

Important Dates

- ~~All Training Data Ready and Released: 4 September 2023~~
- **Evaluation Start: 10 January 2024**
- Evaluation End: 31 January 2024
- Paper Submission Due: 29 February 2024
- Notification to Authors: 1 April 2024
- Camera Ready Due: 22 April 2024
- SemEval-2024 Workshop: TBD, 2024 (co-located with a major NLP conference)

All deadlines are 23:59 UTC-12 ("anywhere on Earth").

Introduction

- It represents a shift in focus from purely textual analysis to a more comprehensive understanding of text that includes numerical data
- Examples include the impact of distinguishing between a 30% and a 3% rise in stock prices on sentiment analysis
- Also affect legal judgments, as seen in the difference between "Stealing 10 dollars" and "Stealing 100,000 dollars
- In clinical contexts, small variations in numerical readings can convey contrasting implications
- Numerically-aware language comprehension and generation

Tasks

Task 1:

- Task focuses on quantitative understanding.
- It comprises three subtasks: Quantitative Prediction (QP), Quantitative Natural Language Inference (QNLI), and Quantitative Question Answering (QQA)
- Uses the Quantitative 101 dataset, which combines publicly available datasets.
- No separate private test in NumEval; encourages collaborative sharing among participants
- For evaluation the Quantitative-101 Score will be used for ranking the overall performance. The Quantitative-101 Score is the average of the macro-F1 score of the QP task and the micro-F1 score of the QNLI and QQA tasks.

Task 2:

- Task involves reading comprehension of numerals in Chinese text
- Models must identify the correct numerical value from four options based on news articles
- NQuAD dataset

News Article: Major banks take the lead in self-discipline. The five major banks' newly-imposed mortgage interest rates climbed to **1.986%** in May. ... Also approaching **2%** integer alert ... Up to **2.5%** ... Also increased by **0.04** percentage points from the previous month ... Prevent the housing market bubble from fully starting.
Question Stem: Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly _____%.
Answer Options: (A) 0.04 (B) 1.986 (C) 2 (D) 2.5
Answer: (C)

Task 3:

- Task focuses on numeral-aware headline generation in English
- Comprises two subtasks: numerical reasoning and headline generation
- Models must compute correct numbers to fill the blanks for news headlines and generate headlines based on provided news
- Using a separate private test set, and for evaluation, must submit the output of the models

News: At least **30** gunmen burst into a drug rehabilitation center in a Mexican border state capital and opened fire, killing **19** men and wounding **four** people, police said. Gunmen also killed **16** people in another drug-plagued northern city. The killings in Chihuahua city and in Ciudad Madero marked one of the bloodiest weeks ever in Mexico and came just weeks after authorities discovered **55** bodies in an abandoned silver mine, presumably victims of the country's drug violence. More than **60** people have died in mass shootings at rehab clinics in a little less than **two** years. Police have said **two** of Mexico's **six** major drug cartels are exploiting the centers to recruit hit men and drug smugglers, ...
Headline (Question): Mexico Gunmen Kill _____
Answer: 35
Annotation: Add(19,16)

Reference Paper for Task 1

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving Numeracy by Input Reframing and Quantitative Pre-Finetuning Task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

- Discusses the issue of innu-meracy in pretrained language models and proposes a solution to improve their understanding of numerals.
- The authors explore the effectiveness of changing notation and pre-finetuning with the comparing-number (CND) task in improving performance in quantitative-related tasks.

Subtask	Question	Answer
QP	FED'S DUDLEY REPEATS EXPECTS GDP GROWTH TO PICK UP IN 2014, FROM [Masked] PCT POST-RECESSION AVERAGE	1
QNLI	S1: Nifty traded above 7500, Trading Calls Today S2: Nifty above 7400	Entailment
QQA	Elliot weighs 180 pounds whereas Leon weighs 120 pounds. Who has a bigger gravity pull? Option1: Elliot Option2: Leon	Option 1

QP - Quantitative Prediction

- This task involves **predicting a numerical value** based on a given input.
- The input can be a natural language sentence or a table.
- The task requires the model to understand the meaning of the input and perform arithmetic operations to arrive at the correct output.
- Here, the authors **fine-tune the BERT and RoBERTa** models on the Numeracy-600K dataset, which is an **eight-class classification task** that includes magnitudes from 1 to 6, decimal, and a magnitude larger than 6.

Format of Numbers is Important

QNLI - Quantitative Natural Language Inference

- This task involves determining whether a given hypothesis is **true, false, or cannot be determined** based on a given premise.
- The **premise and hypothesis** are both natural language sentences that involve numerical information.
- The task requires the model to understand the meaning of the sentences and perform logical reasoning to arrive at the correct answer.
- Here, the authors **fine-tune the BERT and RoBERTa** models on the **QNLI dataset (EQUATE)**, which involves making natural language inferences based on quantitative clues.

QQA - Quantitative Question Answering

- This task involves **answering binary questions** that require numerical reasoning.
- The questions can be in the form of natural language sentences or tables, and the answers are numerical values.
- The task requires the model to understand the meaning of the questions and perform arithmetic operations or logical reasoning to arrive at the correct answer.
- Here, the authors fine-tune the BERT and RoBERTa models on the Task 3 subset of NumGLUE, which is a **binary-classification** dataset that tests the models' ability to understand numerals and semantics.

Focus on Binary Classification

Experimental results

- **Innumeracy Experiment (CND):** The innumeracy phenomenon is observed by testing LMs on the ComNum task using the CND dataset. Two test sets, CND-T1 and CND-T2, with different orders of magnitude, are used to evaluate LM performance. Results show that LM performance drops when tested on CND-T2, indicating the "innumeracy" phenomenon. Input reframing methods (Digit-based and Scientific Notation) mitigate this issue, with Scientific Notation performing the best.

	BERT		RoBERTa		LinkBERT		FinBERT	
	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2
Original	99.86	95.59 (↓ 4.27)	99.44	86.75 (↓ 12.69)	99.92	97.58 (↓ 2.34)	99.55	78.37 (↓ 21.18)
Digit-based	99.96	99.03 (↓ 0.93)	99.92	98.46 (↓ 1.46)	99.99	96.54 (↓ 3.45)	99.96	97.03 (↓ 2.93)
Scientific Notation	99.92	99.68 (↓ **0.24**)	99.82	99.13 (↓ **0.69**)	99.95	99.81 (↓ **0.14**)	99.72	98.78 (↓ **0.94**)

Table 3: Experimental results of ComNum task. The evaluation metric is Micro-average of F1 score (%).

Experimental results

- **Experimental Results (Quantitative Tasks):** Experiments on the quantitative tasks using different LMs (BERT, RoBERTa, LinkBERT, and FinBERT). They compare results with and without input reframing and pre-finetuning.
  - BERT-Based Models: Input reframing and pre-finetuning do not significantly improve the performance of BERT-based models in QP tasks.
  - RoBERTa-Based Models: Input reframing and pre-finetuning significantly improve the overall performance of RoBERTa, making it comparable to BERT-based models. The QNLI task benefits the most from pre-finetuning.
  - FinBERT-Based Models: Proper input reframing improves FinBERT's performance, with the proposed CN-FinBERT outperforming the original FinBERT.

Model	Notation	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
BERT	*Original*	70.44%	57.46%	64.40%	59.20%	72.29%	60.42%	99.91%	53.20%	67.17
	Digit-based	65.38%	54.74%	57.86%	56.46%	71.36%	60.11%	99.11%	**53.75%**	64.85
	Scientific Notation	65.31%	55.99%	**64.42%**	**60.73%**	72.23%	59.66%	99.56%	**53.24%**	66.39
CN-BERT	*Digit-based*	69.93%	54.84%	61.07%	**60.27%**	**75.54%**	**65.39%**	99.42%	52.53%	**67.37**
	Scientific Notation	64.87%	56.40%	**66.39%**	54.70%	**75.41%**	**63.94%**	99.42%	51.90%	66.63
LinkBERT	*Original*	68.81%	55.70%	59.94%	56.85%	73.43%	59.01%	99.91%	54.14%	65.97
	Digit-based	63.76%	55.41%	59.54%	**57.42%**	**73.63%**	**60.17%**	99.73%	53.44%	65.39
	Scientific Notation	65.81%	**56.05%**	57.00%	56.78%	**75.51%**	58.51%	99.82%	**54.33%**	65.48
CN-LinkBERT	*Digit-based*	68.61%	54.44%	**63.59%**	55.08%	71.21%	58.99%	**100.00%**	50.44%	65.30
	Scientific Notation	63.48%	53.15%	**62.02%**	**59.39%**	**75.70%**	**62.61%**	99.73%	52.11%	**66.02**

Model	Notation	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
RoBERTa	*Original*	60.46%	58.03%	60.15%	57.64%	79.58%	58.77%	98.93%	51.96%	65.69
	Digit-based	**69.25%**	57.65%	59.40%	56.69%	78.90%	**62.38%**	**99.91%**	**54.34%**	**67.31**
	Scientific Notation	**64.32%**	55.49%	60.08%	57.41%	78.68%	**60.81%**	**100.00%**	**53.67%**	**66.31**
CN-RoBERTa	*Digit-based*	**64.25%**	55.92%	**68.96%**	**58.80%**	77.99%	**60.99%**	**99.73%**	50.88%	**67.19**
	Scientific Notation	60.28%	54.85%	**62.15%**	**58.74%**	65.92%	**59.59%**	**99.47%**	**52.27%**	64.16

Model	Reframing	QP-Comment
FinBERT	*Original*	65.26%
	Digit-based	**69.89%**
	Scientific Notation	**70.03%**
CN-FinBERT	*Digit-based*	**68.84%**
	Scientific Notation	**69.76%**

Table 6: Results of the FinBERT-based models.

what MathBert Model really is and what can we learn from it?

MathBert is

“ A Pre-Trained Model for Mathematical Formula Understanding”

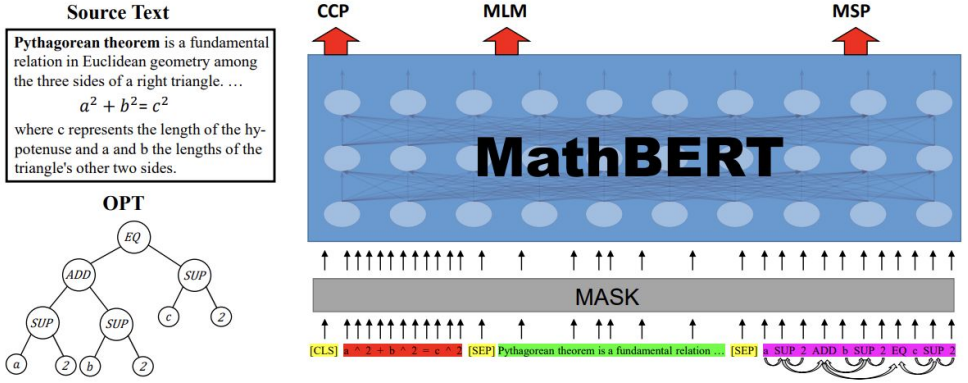
In physics, **mass–energy equivalence** is the relationship between mass and energy in a system’s rest frame, where the two values differ only by a constant and the units of measurement. The principle is described by Albert Einstein's famous formula:

$$E = mc^2$$

The formula defines the energy E of a particle in its rest frame as the product of mass m with the speed of light squared (c^2). Equivalently, the mass of a particle at rest is equal to its energy E divided by the speed of light squared (c^2).

(from Wikipedia)

Figure 1: An example of mathematical formula “$E = mc^2$” with its context, where the text contains rich semantic information of the brief formula.



Why?

like its name suggest : it may not work well with our tasks

→ But that doesn't mean we can't learn from it!

→ what's interesting to me is the input of MathBert model include :

Mathematical formulas represented as LaTeX tokens , which reminds me of

Input Reframing from 2023 winner paper!

→ Maybe, that adds up to the another possible approach we can do this year

```
print(math_df[['sequence', 'token_str', 'score']])
```

```
10 best films of [Num] - What's on your list? 10 best films of [MASK] - What's on your list? 103
                                                sequence   token_str         score
0                  10 best films of - what's on your list?      [MASK]  9.999593e-01
1  10 best films of [unused53] - what's on your l...  [unused53]  1.858963e-05
2                10 best films of 司 - what's on your list?          司  3.048358e-06
3  10 best films of [unused52] - what's on your l...  [unused52]  2.274993e-06
4    10 best films of smiling - what's on your list?     smiling  7.510233e-07
```

What are LaTex Tokens?

- LaTeX tokens refer to the sequence of characters or strings that make up the LaTeX source code representation of a mathematical formula.
- LaTeX is a document preparation system that allows math formulas to be written in a markup language. For example, the formula "x = y^2" would be represented as "$x = y^{2}$" in LaTeX.
- So the LaTeX tokens here would be:
- "$", "x", "=", "y", "^{", "2", "}$"
- These tokens encode the identity of the math symbols and structure of the formula. Using them as input allows models like MathBERT to learn representations of both the surface form (sequences of math symbols) as well as the semantic structure (encoded by the LaTeX math delimiters like ^{ }).
- So in summary, LaTeX tokens provide a symbolic encoding of math formulas that capture both textual and structural information, which MathBERT takes as part of its input representation. The model learns embeddings for these tokens in order to represent the meaning of formulas.

This kind of Input Reframing is nice ,but might not be for us!

- Semantic Gap for Numbers: Representing numbers in LaTeX may introduce a semantic gap, as LaTeX primarily encodes symbolic and structural information. Numerical values often have different semantics than symbols or structures, and encoding them in LaTeX may not capture their inherent meaning effectively.
- Increased Complexity: Representing numbers as LaTeX tokens could introduce unnecessary complexity to the input representation. Numerical values are typically straightforward in their natural form, and representing them using LaTeX may add unnecessary overhead without clear benefits.
- **Task-Specific Requirements**: The success of input reframing depends on the specific requirements of the tasks. For tasks like QP, QNLI, and QQA, the primary focus might be on understanding numerical relationships, comparisons, and answering questions related to quantitative information. Using LaTeX for numbers may not align with these task requirements and could be an unnecessary transformation.
- Data Distribution: Consider the distribution of numerical values in our dataset. If the numerical values play a crucial role in the tasks and have a diverse range, it might be more effective to represent them directly in their natural form rather than encoding them as LaTeX tokens.

so, what did we really learn? Need Task-Oriented Thinking

We need to Explore other Numerical Representations:

- Experiment with different ways of representing numerical information in your model. (e.g., embeddings, context-aware representations).

We need to Utilize Pretrained Language Models:

- Leverage pretrained language models (e.g., BERT, RoBERTa) for contextual understanding of text. Fine-tune these models on your task-specific data to capture the nuances of numerical expressions in the context of natural language.

Data Augmentation:

- Augment your training data to increase the diversity of numerical expressions and their contexts. This can help your model generalize better to unseen examples.

Consider Task-Specific Architectures:

- Depending on the nature of the tasks within NumEval, consider task-specific architectural modifications.

For example, if the tasks involve understanding quantitative comparisons, our model architecture might benefit from mechanisms that capture relational information.

First Step: Inference on Task1

(Re-Creating Paper Work and Learn from It)

BERT Inference (without fine-tuing)

```
[9]  print(training_data[1]['masked'])
     print(training_data[1]['title'])
     filled = fill_mask(training_data[1]['masked'])
```

1s

```
10 best films of [Num] - What's on your list?
10 best films of 2009 - What's on your list?
```

```
df = pd.DataFrame(filled)
print(df[['sequence', 'token_str', 'score']])
```

0s

	sequence	token_str	score
0	10 best films of 2010 - what's on your list?	2010	0.068802
1	10 best films of 2011 - what's on your list?	2011	0.047727
2	10 best films of 2012 - what's on your list?	2012	0.039940
3	10 best films of 2000 - what's on your list?	2000	0.032997
4	10 best films of 2009 - what's on your list?	2009	0.030679

RoBERTa Inference (without fine-tuing)

```
[45] print(training_data[1]['masked'])
     print(training_data[1]['title'])
     filled = fill_mask(training_data[1]['masked'])
```

```
     10 best films of [Num] - What's on your list?
     10 best films of 2009 - What's on your list?
```

```
[44] df = pd.DataFrame(filled)
     print(df[['sequence', 'token_str', 'score']])
```

```
                                                sequence      token_str     score
0      10 best films of  2018 - What's on your list?          2018  0.110871
1       10 best films of ____ - What's on your list?          ____  0.100346
2  10 best films of ------------ - What's on you...  ------------  0.045545
3    10 best films of _______ - What's on your list?       _______  0.043576
4      10 best films of  2017 - What's on your list?          2017  0.033332
```

FinBERT Inference (without fine-tuing)

```
print(training_data[0]['masked'])
print(training_data[0]['title'])
fin_filled = fin_fill_mask(training_data[0]['masked'])
print(fin_filled)
```

```
100 Most Anticipated books releasing in [Num]
100 Most Anticipated books releasing in 2010
[{'score': 0.058827634900808334, 'token': 519, 'token_str': '2008', 'sequence': '100 most
```

```
[157] df = pd.DataFrame(fin_filled)
print(df[['sequence', 'token_str', 'score']])
```

	sequence	token_str	score
0	100 most anticipated books releasing in 2008	2008	0.058828
1	100 most anticipated books releasing in 2006	2006	0.046661
2	100 most anticipated books releasing in 2007	2007	0.041990
3	100 most anticipated books releasing in the	the	0.039691
4	100 most anticipated books releasing in q3	q3	0.038643

LinkBERT Inference (without fine-tuing)

```
[151] print(training_data[0]['masked'])
      print(training_data[0]['title'])
      lin_filled = lin_fill_mask(training_data[0]['masked'])
      print(lin_filled)
```

```
100 Most Anticipated books releasing in [Num]
100 Most Anticipated books releasing in 2010
[{'score': 0.02094154804944992, 'token': 7342, 'token_str': 'Anglo', 'sequence': '100 Most Anticipated boo
```

```
df = pd.DataFrame(lin_filled)
print(df[['sequence', 'token_str', 'score']])
```

	sequence	token_str	score
0	100 Most Anticipated books releasing in Anglo	Anglo	0.020942
1	100 Most Anticipated books releasing in figure	figure	0.007586
2	100 Most Anticipated books releasing in mess	mess	0.005004
3	100 Most Anticipated books releasing in phase	phase	0.004986
4	100 Most Anticipated books releasing in certain	certain	0.004460

To Be Continued

1. Task 1-2 and Task 1-3
2. Model Recreation
3. Model Fine-Tuing