Exploiting Numerical-Contextual Knowledge to Improve Numerical Reasoning in Question Answering

Jeonghwan Kim **Junmo Kang*** **Kyung-min Kim***
Giwon Hong* **Sung-Hyon Myaeng**
School of Computing, KAIST
Daejeon, Republic of Korea
{jeonghwankim123, jumon.kang, kimdarwin,
giwon.hong, myaeng}@kaist.ac.kr

Abstract

Numerical reasoning over text is a challenging subtask in question answering (QA) that requires both the understanding of texts and numbers. However, existing language models in these numerical reasoning QA models tend to overly rely on the pre-existing parametric knowledge at inference time, which commonly causes hallucination in interpreting numbers. Our work proposes a novel attention masked reasoning model, the **NC-BERT**, that learns to leverage the number-related contextual knowledge to alleviate the over-reliance on parametric knowledge and enhance the numerical reasoning capabilities of the QA model. The empirical results suggest that understanding of numbers in their context by reducing the parametric knowledge influence, and refining numerical information in the number embeddings lead to improved numerical reasoning accuracy and performance in DROP, a numerical QA dataset.

1 Introduction

Understanding numbers in text is critical when dealing with numerical reasoning problems over text. Most previous works (Ran et al., 2019; Andor et al., 2019; Chen et al., 2020b; Gupta et al., 2019; Chen et al., 2020a; Geva et al., 2020; Saha et al., 2021) on such numerical reasoning over text have shown substantial amounts of performance gain in numerical question answering (QA) tasks such as DROP (Dua et al., 2019). While these models display stellar performance, previous studies that evaluate model's numerical reasoning robustness (Talmor et al., 2020; Kim et al., 2021; Al-Negheimish et al., 2021) suggest that these QA models that typically depend on large language models (LM) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) suffer from a limitation: the disregard of context information at inference time for numerical reasoning.

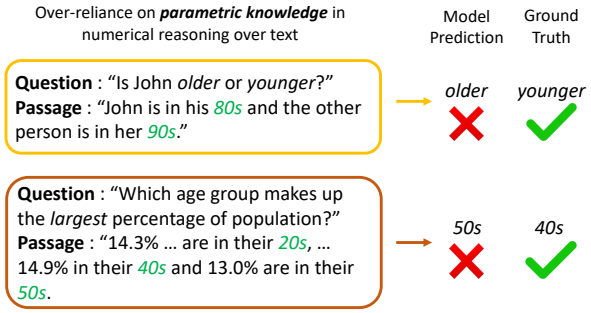

Figure 1: Cases that testify the over-reliance on parametric knowledge by language models like BERT during numerical reasoning. These attest to the language models' lack of ability to properly understand numbers in their given context.

This issue is exemplified in Figure 1 showing a numerical reasoning question (Talmor et al., 2020), where LMs fail to decouple number values from their types; a number is understood as in the LM, not as the value to be interpreted in the local context. For example, in the first case of Figure 1, the model misinterprets the numbers "80s" and "90s" as YEAR-type from the LM, instead of AGE-type obtainable from the given local context, leading to an incorrect answer for the "older" relation. Had the model leveraged the context of the passage instead of pre-existing knowledge reflected in the LM's parameters, the model should have correctly predicted the answer. Such knowledge is commonly referred to as *parametric knowledge*, which is learned from the training instances that the model has encountered prior to the inference process as in the above example. Previous works (Krishna et al., 2021; Bender et al., 2021; Al-Negheimish et al., 2021) testify that such propensity to rely on previously learned knowledge instead of looking at the present context information frequently haunts large LMs. As such, we hypothesize that the capability to override the parametric knowledge captured in the number embeddings with the relevant *contex-*

*Equal contribution.

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1811 - 1821

tual knowledge is crucial to properly understand the numbers at inference time, and thus perform accurate and robust numerical reasoning.

To override the detrimental influence of parametric knowledge on model's numerical reasoning capability, our work proposes a novel attention masking scheme. This **N**umerical-**C**ontextual attention mask (**NC-Mask**) explicitly induces the number-related flow of contextual information into the number embeddings to enable the model to properly interpret the numbers according to the context given, and thereby improve the model's numerical reasoning capability. This attention mask is predicated upon the following two intuitions: (i) numbers are always related to entities in the same sentence and (ii) a number type (e.g., YEAR, AGE, QUANTITY) is defined by its surrounding words. While such *entity-number* and *type-number* interactions rely on the self-attention mechanism of the Transformer architecture (Vaswani et al., 2017), the local relations to be captured from the context are not always implicitly captured by the LM's self-attention. Our **NC-Mask** is intended to consolidate such relations.

On the other hand, this "overwriting" of parametric knowledge in the number embeddings can cause the characteristic numeracy information (e.g., magnitude) in the embeddings to be diluted. Previous works (Wallace et al., 2019; Sundararaman et al., 2020; Pal and Baral, 2021; Kim et al., 2021) show that LMs possess, to a certain degree, the notion of numeracy in their parameters. For example, the information that number 2 is less than or equal to 9 ($2 \leq 9$) is contained within the embeddings of those numbers. In order to avoid losing such valuable numeracy information, we adopt the DICE regularization (Sundararaman et al., 2020), a magnitude-inducing regularizer to instill relative magnitude hierarchy into the number representations and thus replenish the diluted numeracy in the embeddings for numerical reasoning.

The empirical results of this work suggest that our attention masking strategy improves the numerical reasoning capability of the QA model. In Section 5, we also provide a detailed analysis on the optimal application of our masking strategy on the different layers and heads of the QA model's encoder. Our results also imply that our attention masking allows extra layers that leverage the knowledge instilled by the masked channels to be added, leading to the scaling up of the model without pre-training the extra layers.

2 Related Work

2.1 Parametric Knowledge vs. Contextual Knowledge

Downstream NLP tasks often require the use of two disparate sources of knowledge: parametric and contextual knowledge. Previous works (Longpre et al., 2021) reveal that conflicts between the two types of knowledge occur from over-reliance on parametric knowledge, which is exacerbated by the significant overlap between the passage-question pairs in the training and validation sets (Krishna et al., 2021; Al-Negheimish et al., 2021). For the task of numerical QA, a related work (Talmor et al., 2020) reveals that models fail to understand numbers in the given passage because they rely on the parametric knowledge (i.e., memorization) within the pre-trained number embeddings. Such lack of contextual understanding of numbers prevent these models from properly interpreting numbers and thus inhibit effective numerical reasoning over text.

2.2 Numeracy in Language Models

Recent studies on the numeracy of large LMs reveal that number embeddings constructed by either the non-contextual embedding methods (Sundararaman et al., 2020) or large language models (Wallace et al., 2019; Talmor et al., 2020; Sundararaman et al., 2020; Kim et al., 2021) possess, to a certain degree, a prior notion of numeracy such as magnitude. However, the numeracy in the LMs are neither deterministic nor accurate (Wallace et al., 2019) like the scalar numbers. These representations, furthermore, require additional refinement measures to induce additional numeracy (Sundararaman et al., 2020). Such lack of numeracy can lead to a few of the following problems: (i) confusing numbers of similar magnitude (Talmor et al., 2020), and (ii) calculating wrong numerical answers (Geva et al., 2020). Since our masking scheme can bring about such issues due to diluted numeracy in the number embeddings, we adopt a numeracy-inducing regularization term from Sundararaman et al. (2020) to alleviate this problem and improve the overall accuracy in numerical calculations.

3 Approach

In this work, we propose an attention masked question answering (QA) model, namely the **N**umerical-**C**ontextual BERT (**NC-BERT**), that learns to rely on the contextual knowledge and exhibits improved numerical reasoning capability in QA. First, we construct a tripartite attention mask (i.e., **NC-Mask**) over a self-attention layer on top of the encoder to direct the flow of necessary contextual information to the number embeddings, so that number-related contextual knowledge such as entity-number and type-number relations are effectively leveraged during model training. We then adopt a complementary numeracy-inducing regularizer to counteract the numeracy dilution issue caused by our masking strategy and to further enhance the accuracy of numerical calculation.

3.1 Preliminaries

We first provide a qualitative analysis on an LM's number token embeddings to identify and reveal the type of pre-existing parametric knowledge in them, which induces model dependence on parametric instead of contextual knowledge. To this end, we first sample all the number embeddings from a pre-trained BERT-base model (Devlin et al., 2019). Then, we use an off-the-shelf similarity search tool, FAISS (Johnson et al., 2017), to calculate the cosine similarity between each number embedding and every other non-number token embeddings in BERT's vocabulary. The next step is to sample tokens with top-k ($k = 5$) similarity scores for each number token. Note that we exclude other numeral tokens (e.g., "two", "71"), non-alphabetical tokens, and special tokens such as [PAD] and [SEP] as seen in Table 1. However, we retain number-related tokens such as "15th" and "1990s" as indicators of DATE-related parametric knowledge within the embeddings.

In Table 1, we can see that the number embeddings are deemed to contain semantics about date, month or other quantity information. Furthermore, numbers like "2018" already contain DATE-related information, which does not seem out-of-context considering that "2018" is more often than not used in such contexts. However, this knowledge influences model decisions negatively when questions use "2018" for a non-date scenario, resulting in an error as in Figure 1. Numbers like "50" and "114" that are seldom used in DATE context also hold such DATE information, further suggesting

What is in Number Embeddings	
`2018`	currently, october, 1990s, july, 19th
`50`	1950s, various, significant, many, substantial
`114`	12th, 11th, 14th, 15th, 13th
`2`	ii, several, various, 4th, iii
`11`	11th, 12th, 10th, 8th, 13th

Table 1: The leftmost column contains the number tokens from BERT (`bert-base-uncased`)'s vocabulary and the rightmost column contains the top-5 similarity-scored tokens that correspond to each number token.

that relying entirely on such parametric knowledge induces models to make a wrong prediction. It is evident, therefore, we need to pay attention to the distinction and interplay between the *parametric knowledge* identified in this analysis and *context-specific semantics* in QA models during numerical reasoning.

3.2 Base Model Architecture

A QA model for numerical reasoning needs to conduct both the span extraction from text and numerical answer generation. This steered our work to leverage GenBERT (Geva et al., 2020) as our baseline. Given a question (**q**) with m tokens and a passage (**p**) with n tokens, we construct an input sequence that consists of the BERT special tokens as follows: <`[CLS]` $q_1, q_2, ..., q_m$, `[SEP]`, $p_1, p_2, ..., p_n$>. Then, we produce the contextualized representations, **M**, for each token after passing the input sequence through a pre-trained LM encoder of choice (in this case, BERT).

$$\mathbf{M} = Encoder(\mathbf{q}, \mathbf{p}) \quad (1)$$

The representation **M** is then passed on to two different kind of prediction modules for answer prediction; namely, span extraction and decoder modules. The span extraction module, $\mathbf{H}_{span}$, calculates the probabilities of start and end spans, whereas the decoder module, $\mathbf{H}_{dec}$, generates answers that are not found within the passage but can only be calculated by numerical reasoning such as addition and counting.

$$\mathbf{H}_{span} : (\hat{y}_{start}, \hat{y}_{end}) = \arg\max_{s<=e} P(\mathbf{M}_s)P(\mathbf{M}_e) \quad (2)$$

$$\mathbf{H}_{dec} : \hat{y}_i = \arg\max_{i} P(y_i|y_1, y_2, ..., y_{i-1}, \mathbf{M}) \quad (3)$$

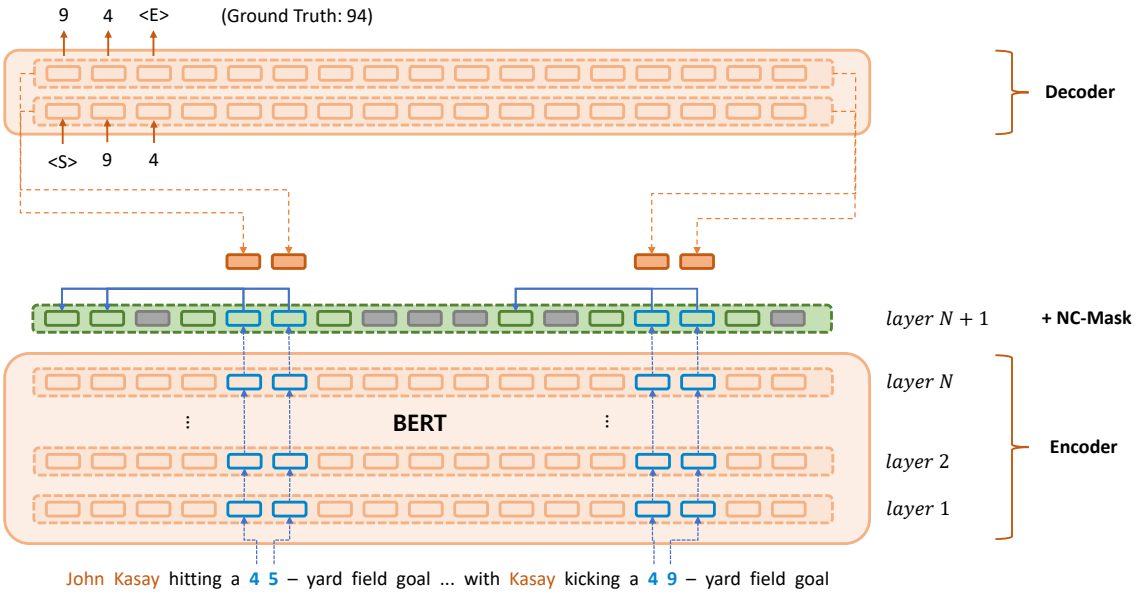

Figure 2: A visual representation of Numerical-Contextual BERT (**NC-BERT**). The green layer represents the extra layer with **NC-Mask** on top of the encoder. The blue-highlighted path represents the digits that appear in the passage, along with the entity-number channel that is highlighted in blue in the **NC-Mask** layer. The decoder-number channel is presented in the upper half of the figure. Here, we omit the type-number channel for ease of understanding.

In the above equation, $\mathbf{M}_s$ and $\mathbf{M}_e$ denote the start and end token representations, respectively. We then train the model by minimizing the loss, L_{ans}, which is a marginal probability over the output of two modules as follows:

$$L_{ans} = -log(\mathbf{p_{dec}} \cdot \prod_{s=1}^{S} p(y_s) + \sum_{\mathbf{h} \in \mathbf{q,c}} \mathbf{p_h} \cdot \sum_{(i,j) \in T} p_h(i,j) \quad (4)$$

where $\mathbf{p_{dec}}$ and $\mathbf{p_h}$ are the module type probabilities from a single layer feed-forward network (i.e., the type module). S is the length of the answer sequence generated by the decoder, and T is the set of all possible answer spans from the passage. We omit the conditionals for brevity.

3.3 NC-BERT for Numerical Reasoning

We describe NC-BERT, an encoder-decoder model designed for numerical reasoning, with the intuition and mechanism behind the three parts of the **NC-Mask** scheme. Also explained in this section is the rationale for the numeracy-inducing regularization term to deal with the numeracy dilution issue caused by the attention masking scheme. We fine-tune the model with our mask to enable the model to use the mask as a medium to effectively aggregate numerical-contextual knowledge from texts.

3.3.1 NC-Mask

We construct an attention mask that allows the number-related contextual information to be channeled to the number embeddings. Two main intuitions are: (i) numbers are bound to the entities in the same sentence, and (ii) the words surrounding a number define its type. In order to reflect the intuitions, we construct two types of attention masks referred to as *entity-number* and *type-number* channels, which attempt to leverage the number-related input context and adjust the influence of parametric knowledge in the number embeddings. In addition, we devise the third channel, the *decoder-number* channel for two reasons: (i) the decoder needs only to "calculate" number sequence answers and (ii) the non-essential passage tokens act as noise during numerical calculation of the decoder.

Entity-Number Channel To construct the entity-number channel, we first extract every entity[1] and digit in each sentence. For each sentence, we construct an entity index set, E, and a dictionary with the digit indices as keys and E as values. E' (where $E' \subseteq E$) is assigned as a value to a digit

[1]Using Stanford Stanza toolkit for NLP (Qi et al., 2020)

index if and only if the corresponding digit and E' belong in the same sentence. The dictionary is then used to construct the attention mask, A_E, as follows:

$$\alpha = softmax(A_E \odot \frac{QK^T}{\sqrt{d_k}})V \quad (5)$$

where α is the normalized attention score and A_E is the attention mask that drowns out the tokens that are irrelevant to interpreting numbers in the text, namely, the non-entity tokens. To elaborate, when the query, q_i, of the i^{th} number token attends to all the $m + n$ other sequence of tokens in the text during self-attention, A_E leaves the attention scores from the i^{th} number token to the entities in the same sentence unchanged, while zeroing out all the other attention scores to preclude their corresponding values ($v \in V$) from being added to the subsequent-layer representation of the i^{th} number token.

Type-Number Channel Type-number channel is constructed in a similar fashion as in Equation 5 by creating another attention mask, A_W. We first define a window of size k for every number. Within each window, we select the m (where $m/2 <= k$) immediately neighboring words and construct another dictionary with the digit indices as keys and their neighboring word indices as values. Similar to A_E, A_W is constructed to block out all the non-essential, noisy interactions from the number embeddings to irrelevant tokens in the text, while retaining the attention to the m immediately neighboring words that define the number type.

Decoder-Number Channel In a regular encoder-decoder Transformer architecture, the decoder uses its query, Q_{dec}, to attend over the value embeddings, V_{enc}, of the encoder's last hidden states (Vaswani et al., 2017). In this work, the decoder employs a new type of source attention mask, A_{src}, along with the above two masks, that confines the decoder's query to attend only over the numbers and the question tokens of the encoder's last hidden states (Equation 6).

$$\gamma = softmax(A_{src} \odot \frac{Q_{dec}K_{enc}^T}{\sqrt{d_k}})V_{enc} \quad (6)$$

where γ is the source attention score. Since the number embedding is constructed to contain the number-related contextual knowledge, the decoder can learn to attend to the numbers by utilizing such knowledge within the number embeddings and perform calculations.

Figure 2 shows the overall framework, **NC-BERT**, with the NC-Mask applied on top of the encoder and decoder-number channel selectively attending to the last hidden state representations of the number tokens. The details of where the NC-Mask is applied in the encoder and the reason thereof is elaborated in Section 5.4

3.3.2 Numeracy-Inducing Regularization

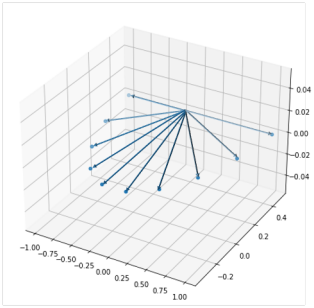

Figure 3: A visual depiction of how the **DICE-loss** induces the relative magnitude hierarchy among the digit embeddings. The leftmost arrow depicts the embedding of 0 and the rightmost the embedding of 9.

Overwriting the parametric knowledge with **NC-Mask**, however, can cause pre-existing numeracy characteristic like the magnitude of a number, to be erased from the number representations. To alleviate such dilution of numeracy and to revive the magnitude characteristic of numbers in the representations, we adopt the numeracy-inducing regularization (**DICE-loss** in short) term (Sundararaman et al., 2020). It samples two random digits, a and b, from the given input text and their corresponding hidden states, $\mathbf{v_a}$ and $\mathbf{v_b}$, from the last layer of the encoder. Then, it calculates the difference between the scalar distance of a and b and the cosine distance, d_{cos}, of their corresponding last layer representations as follows.

$$L_{num} = \|2\frac{|a-b|}{|a|+|b|} - d_{cos}(\mathbf{v_a}, \mathbf{v_b})\|_2 \quad (7)$$

DICE-loss (Figure 3) acts as an effective regularizer to induce the relative magnitude relation among the digits, thereby adjusting model parameters to generate contextualized number embeddings reflecting such hierarchy of magnitude. With the **DICE-loss**, the final expression for the objective ends up as follows: $L = L_{ans} + L_{num}$.

4 Experiments

To validate the proposed approach, we establish the two following research questions:

Q1. "Does number-related contextual knowledge help improve numerical reasoning?"

Q2. "Does compensating for the diluted numeracy give more accurate numerical calculation?"

Q1 is assessed by the channel-wise application of the **NC-Mask** on our model encoder. By evaluating each channel and its effect on the model's reasoning capability, we reveal how the number-related contextual knowledge influence the baseline's numerical reasoning. We also determine the optimal placement of the **NC-Mask** by exploring where in the encoder should the mask be applied for better numerical reasoning. Q2 is evaluated by applying **DICE-loss** and thereby replenishing the diluted numeracy of the number embeddings. In addition, we experiment with the least significant digit first (LSDF) generation scheme on our encoder-decoder architecture. The goal is to explore the idea of taking the "carry" into account to bring about an additional benefit to the overall numerical reasoning capability of our model (details in the Appendix).

4.1 Dataset

DROP (Dua et al., 2019) is a numerical reasoning over text dataset for QA models. This evaluation dataset consists of a total of 9,536 question-answer pairs with respect to 582 passages. The answer types are largely: Number, Date and Others, where Others refer to span type answers.

4.2 Baselines

For the baseline, we use GenBERT (Geva et al., 2020), a Transformer encoder-decoder model initialized with BERT-base parameters pre-trained with simple arithmetic and textual number reasoning tasks. On top of the baseline, we empirically evaluate the effectiveness of (i) **NC-Mask** (ii) the numeracy-inducing regularization (**DICE-loss**), along with the LSDF generation technique for additional reasoning enhancement. We do not incorporate other state-of-the-art models like QDGAT (Chen et al., 2020a) and NumNet (Ran et al., 2019), since they disregard the numeracy understanding part (Wallace et al., 2019) by simply employing specialized heads that learn to assign {-1, 0, +1} on numbers for summation, and they do not actually perform implicit calculation to derive numerical answers by delegating the calculation part to a symbolic calculator.

4.2.1 Implementation Details

The model is based on GenBERT and is trained using RTX3090 NVIDIA GPU. With the training batch size of 16, the hidden size of 768, the learning rate of 3e-5 and an Adam optimizer with a linear warm up of 0.1. The rest of the hyperparameters are in the Appendix.

5 Results

In this section, we explain the results of our **NC-Mask** scheme and complementary numeracy regularizer by comparing the per answer-type Exact Match (EM) and F1 scores. The results also include head- and layer-wise probing done to determine the optimal masking position in the model.

5.1 Leveraging Contextual Knowledge

As in Table 2, the addition of **NC-Mask** leads to noticeable performance improvements over the baseline. The Entity-Number channel proves to be the largest benefactor to the model's enhanced numerical reasoning capability. Such result can be interpreted from the fact that numbers now share high semantic similarity with the entities in question, which in turn improves the model's numerical reasoning by incorporating those numbers during the calculation. The Type-Number channel, in contrast, turns out to contribute most to the **Date**-type questions; the result is likely caused by the increased interaction between the Date-related tokens and numbers. For the Decoder-Number channel, through the ablation study in Table 2, we prove that the channel is a necessary component of the **NC-Mask** scheme, given the performance degradation in both the **Number** and **Date** type questions in its absence.

5.2 Counteracting Numeracy Dilution

With **NC-Mask** amplifying the influence of contextual knowledge in number embeddings, we now deal with the numeracy dilution issue. The results in Table 3 show substantial improvement in numerical reasoning performance. We also test **DICE-loss** after removing **NC-Mask** to empirically prove that the dilution of numeracy by our masking strategy indeed adversely affects model performance and requires the regularization. When adding the regularizer to the baseline alone (DICE (w/o **NC-Mask**)), we see a drop in performance, meaning that employing the regularizer is ineffective considering the pre-existing numeracy. On the contrary, we evidence a major increase in performance by applying

Model	Number		Date		Others		All	
	EM	F1	EM	F1	EM	F1	EM	F1
GenBERT	73.05	75.21	52.44	56.37	70.17	74.53	68.75	72.30
+ Entity-Number	73.37	76.24	52.41	56.33	71.02	**75.98**	69.10	72.61
+ Type-Number	73.09	75.30	**53.60**	**56.59**	70.15	74.34	68.82	72.31
+ Decoder-Number	73.10	75.37	52.40	55.98	70.33	74.82	68.78	72.34
NC-Mask	**74.16**	**76.89**	53.27	56.32	**71.24**	75.10	**69.17**	**72.65**
- Decoder-Number	73.74	76.33	53.25	56.31	71.24	75.10	69.09	72.40

Table 2: Evaluation on the DROP evaluation dataset. The first half shows the results per answer type for independently applying the three channels of **NC-Mask**, in order to analyze the individual contextual knowledge influence on the model's numerical reasoning performance. **NC-Mask** provides the results of all the channels combined, with an extra ablation result on the decoder-number channel.

Model	Number		Date		Others		All	
	EM	F1	EM	F1	EM	F1	EM	F1
NC-Mask	74.16	76.89	53.27	56.32	71.24	75.10	69.17	72.65
+ DICE (w/ **NC-Mask**)	75.03	77.70	**53.42**	**56.45**	72.10	75.63	69.93	73.55
+ DICE (w/o **NC-Mask**)	73.36	76.12	52.42	55.98	70.17	74.40	68.87	72.38
NC-BERT	**75.09**	**77.72**	53.41	56.31	**72.12**	**75.68**	**69.96**	**73.59**

Table 3: Results of **NC-Mask** in coordination with **DICE-loss** to alleviate the numeracy dilution issue. **NC-BERT** incorporates the LSDF generation scheme.

NC-Mask and **DICE-loss** together. The implications of this outcome are: (i) numeracy dilution occurs when overriding parametric knowledge in the embeddings and can be addressed by adopting the regularization term, and (ii) the regularization and our masking strategy are complementary.

5.3 Head Replacement with NC-Mask

Heads and Layers		EM	F1
Original		68.75	72.30
All Layers & Heads		37.53	40.77
Last Layer	All-Heads	68.83	72.35
	Single-Head	**69.12**	**72.88**
	Odd-Heads	68.76	72.19
	Even-Heads	68.74	72.11

Table 4: Results after applying **NC-Mask** to the encoder. "All" refers to applying **NC-Mask** to every head and layer in the encoder. Single, Odd and Even-Heads refer to the last layer head masking with **NC-Mask**.

To determine which part of the encoder should our masking scheme be applied to, we thoroughly investigated the effects of **NC-Mask** on the heads and layers of the encoder. The result is shown in Table 4, where "Original" is the original performance of our baseline under our setting.

We first applied **NC-Mask** to all the heads and layers (All) of the encoder. Our initial assumption was that if a model could simply learn to attend to useful, number-related context, the model would easily leverage such information for more accurate numerical reasoning. However, the performance drops drastically (-31.53 in F1), suggesting that neglecting the roles of heads and layers of the encoder is detrimental to numerical reasoning over text. This result is also in correspondence with previous works (Rogers et al., 2020; Jo and Myaeng, 2020), where the roles of heads and layers are already defined during the pre-training of the model.

Based on the result, we then applied **NC-Mask** to the heads of the encoder's last layer since the last hidden states are the ones used by the decoder and span extraction module to generate the answer. The result implies that imbuing **NC-Mask** to induce the learning of number-related context is important. However, as the results of "Single-Head" and the other heads in the last layer suggest that considering each head's role is critical to model acquiring such contextual knowledge beneficial to its numerical reasoning capability.

Number of Heads (k)	EM	F1
No New Layer	68.75	72.30
0	68.54	72.11
1	68.68	72.27
2	69.22	72.87
4	69.21	72.86
6	69.52	73.00
12	**69.96**	**73.59**

Table 5: Model performance after applying NC-Mask on k different heads of the layer on top of the encoder of **NC-BERT**. $k = 0$ means a vanilla Transformer encoder layer without the NC-Mask.

5.4 The Extra Layer for Masked Interaction

Applying **NC-Mask** to the last layer replaces one of the pre-trained heads. Here, we see that the loss of encoder's original linguistic capability is inevitable to acquire the additional numerical reasoning skill. Thus, to retain the original textual reasoning capability of the encoder and provide additional numerical-contextual knowledge, we add an extra layer with **NC-Mask** on top of the encoder. Here, we experiment with the number of heads (k) in an effort to figure out the optimal number of heads the model needs to acquire the most accurate numerical reasoning capability.

Our initial assumption was twofold: (i) just like in the last layer's case, a single head would result in the best performance, or (ii) there would be some "sweet spot" in which the model shows the optimal performance. On the contrary to our expectations, the results shown in Table 5 display an entirely different pattern from the ones in Table 4.

From the result of $k = 0$, we confirm that the improvement in model performance does not arise from the naïve addition of an extra self-attention layer, since it rather exhibits a drop in performance. With the incremental addition of the **NC-Mask**, we evidence a proportional increase in performance, which reaches the peak when the maximum number of heads ($k = 12$) is masked. Our interpretation to such disparity in masking patterns is that the extra self-attention layer is not a pre-trained layer unlike the last layer of the encoder; meaning, while the heads of the last layer have pre-defined roles, the extra layer is a randomly initialized parameter with no head-wise roles defined. This implies that the **NC-Mask**ed layer acted as a single, giant "head" that induced useful numerical-contextual knowledge into the encoder, which in turn improved the model's numerical reasoning capability.

Model	Entity	Type	Other
Numbers (Original)	21.61	35.83	42.56
Numbers (**NC-BERT**)	47.14	52.86	-

Table 6: Changes (in %) in the attention patterns from the numbers in passages to different relation types within the encoder. **NC-BERT** exhibits amplified attention magnitude in all the three relation types compared to its original counterpart.

Recall@K	GenBERT	**NC-BERT**
1	0.0049	**0.0081**
2	0.0103	**0.0171**
5	0.0295	**0.0434**
10	0.0689	**0.0840**
20	0.1492	**0.1513**
50	**0.2691**	0.2340

Table 7: Evaluation result of the Recall@K from the number to entity and type-defining words. The entity and type-defining tokens are treated as the ground truth labels, and the K represents the number of top-K retrieved tokens using the corresponding number embeddings.

5.5 Interaction Between Numbers and Contextual Knowledge

In Table 6, we provide an attention pattern analysis to show that with our masking scheme, the number representations readily acquire the relevant contextual knowledge. The numbers shown in the table represent the attention scores from the number tokens in the passage to entity tokens, number-type tokens, and all the other tokens in the passage. For Numbers (Original), we have normalized the attention scores over all the heads in the last layer of the encoder, whereas the Numbers (**NC-BERT**) list the normalized attention scores over all the heads of the extra layer with the **NC-Mask**, where every head ($k = 12$) is masked as in our final model architecture. The results testify that the masking scheme successfully increases the amount of interaction between numbers and their related contextual knowledge (e.g., entity and type-defining words), which in turn led to the improved reasoning accuracy of our model.

On top of the increased interaction between numbers and related contextual knowledge, in Table 7,

Question&Answer	Passage	GenBERT	NC-BERT
Q: How many active military personnel and reserve are in the Croatian Armed Forces?	The total number of active military personnel in the Croatian Armed Forces stands at 14,506 and 6,000 reserves working in various service branches of the armed forces. In May 2016, Armed Forces had ...	14,506 + ? = 14,506	14,506 + 6,000 = 20,506
Q: How many more Macedonians were there compared to Albanians according to the 2002 census?	Skopje, as the Republic of Macedonia as a whole, is characterised by a large ethnic diversity ... According to the 2002 census, Macedonians were the largest ethnic group in Skopje, with 338,358 inhabitants, or 66.75% of the population. Then came Albanians with 103,891 inhabitants (20.49%), ...	338,358 - ? = 328,337	338,358 - 103,891 = 234,467

Table 8: The case study of the DROP dataset. The table juxtaposes the model prediction cases for GenBERT and NC-BERT to qualitatively illustrate the differences between the reasoning of the two models. The calculated answers indicate that NC-BERT is able to relate numbers with their corresponding entities and type-words, leading to improved numerical reasoning accuracy.

we provide the cosine similarity retrieval result of entity and type-defining output representations using the number embeddings in terms of Recall@K. With the two kinds of contextual knowledge as the ground truth tokens, we calculate the recall for the top-K tokens retrieved by the number embeddings in the encoder output. The result in Table 7 shows the increased similarity between the numbers and the corresponding entity and type representations, suggesting that our masking scheme successfully increases the similarity and thus the interaction between useful contextual knowledge and the numbers in text.

5.6 Qualitative Study

For an intuitive understanding of the effect of our proposed method, we provide a case study on the DROP dataset in Table 8. As the cases suggest, our model is better able to relate numbers in text with their pertinent entities (e.g., Croatian Armed Forces) and type words (e.g., reserves), which serve the model with useful entity-number and type-number information that lead to improved numerical reasoning capability of our model. In contrast, the baseline fails to relate the numbers needed for the calculation, which results in wrong answers.

6 Conclusion

This work proposes a novel attention masking scheme, **NC-Mask**, to relieve question answering (QA) models of the language model (LM) encoder's over-reliance on parametric knowledge and improve the numerical reasoning accuracy and robustness. Our analyses and empirical results provide strong evidence that BERT, a commonly used encoder in QA models, needs to employ the extra attention channels to leverage the number-related contextual knowledge for robust numerical reasoning instead of entirely relying on the inherent self-attention mechanism. By additionally adopting a numeracy-inducing regularization term, our work also shows that the proposed masking scheme and regularization are complementary, and retaining numeracy is essential for accurate numerical calculation. Future efforts should focus on increasing the scale of models with the masking scheme, since masked, attention-constrained layers appear to more positively contribute to model's reasoning capability than the addition of fully self-attentive layers.

Acknowledgements

This work was supported by Institute for Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2013-2-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services) and (No. 2021-0-02155, Developing Context/Number Embedding based Numerical Reasoning).

References

Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. 2021. Numerical reasoning in machine reading comprehension tasks: are we there yet? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9643–9649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations

and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5949–5954.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020a. Question directed graph attention network for numerical reasoning over text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6759–6768.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020b. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. In *International Conference on Learning Representations*.

Jae-young Jo and Sung-Hyon Myaeng. 2020. Roles and utilization of attention heads in transformer-based neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3404–3417, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Jeonghwan Kim, Giwon Hong, Kyung-min Kim, Junmo Kang, and Sung-Hyon Myaeng. 2021. Have you seen that number? investigating extrapolation in question answering models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7031–7037, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kuntal Kumar Pal and Chitta Baral. 2021. Investigating numeracy learning ability of a text-to-text transfer model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3095–3101, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. Numnet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Amrita Saha, Shafiq Joty, and Steven CH Hoi. 2021. Weakly supervised neuro-symbolic module networks for numerical reasoning. *arXiv preprint arXiv:2101.11802*.

Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for numeracy-preserving word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4742–4753, Online. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters of the Model

In this section, we provide the important hyperparameters used during the training of our model.

Hyperparameter	GenBERT	**NC-BERT**
Batch size	16	16
Hidden size	768	768
Max. Sequence length	512	512
Learning rate	3e-5	3e-5
Optimizer	AdamW	AdamW
Seed	42	42
Approx. runtime	50 hrs	51.5 hrs

Table 9: Hyperparameters of the two models of this work. The hyperparameters are set so the two models are on an equal footing for fair comparison.

A.2 LSDF: Least Significant Digit First Generation

Our final model, **NC-BERT**, employs the least significant digit first (LSDF) generation scheme during the fine-tuning of the model. The intuition behind the LSDF generation is simple considering the basic rules of addition. When the digits of lower significance add up with their sum greater than or equal to 10, then a carry of 1 occurs which is then passed on to the subsequent significant digit for addition. Such sequence of carries can only happen when the values of lower significance add up in advance. LSDF incorporates this elementary arithmetic rule to the generation of number answers, simply by reversing the order of the number sequence answer (e.g., 127 $\rightarrow$ 721). This digit-position reversing acts as an additional schematic alteration to our encoder-decoder generative architecture, which turns out to benefit the numerical reasoning capability of the model slightly. Our results also imply that considering the intuitive arithmetic calculation steps is important in numerical reasoning.