Numerical Reasoning in Question Answering

Department of Computer Science, Yonsei University

Seungone Kim

louisdebroglie@yonsei.ac.kr

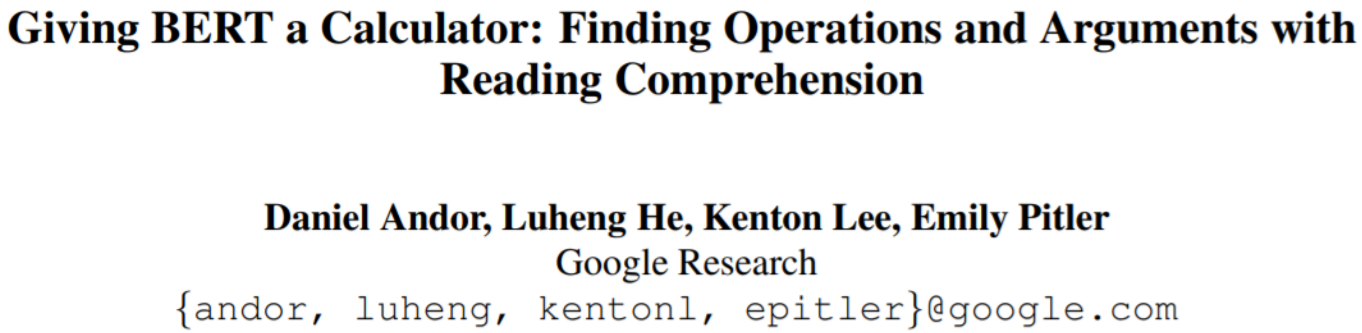
Referenced Papers

- Prerequisites / Additional Papers
  - DROP : A reading comprehension benchmark requiring discrete reasoning over paragraphs [NAACL 2019]
  - QANet : Combining local convolution with global self-attention for reading comprehension [ICLR 2018]

  - Solving general arithmetic word problems [EMNLP 2015]
  - Deeper insights into graph convolutional networks for semi-supervised learning [AAAI 2018]

  - Do nlp models know numbers? Probing numerancy in embeddings [EMNLP 2019]
  - A multi-type multi-span network for reading comprehension that requires discrete reasoning [EMNLP 2019] => NEXT WEEK!
  - Learning to solve arithmetic word problems with verb categorization [EMNLP 2014]
  - MAWPS : A math word problem repository [ACL 2016]
  - Neural symbolic reader : Scalable integration of distributed and symbolic representations for reading comprehension [ICLR 2020]

- Key Papers
  - Giving BERT a Calculator : Finding Operations and Arguments with Reading Comprehension [EMNLP-IJCNLP 2019]
  - NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP-IJCNLP 2019]
  - Injecting Numerical Reasoning Skills into Language Models [ACL 2020]



Giving BERT a Calculator: Finding Operations and Arguments with Reading Comprehension

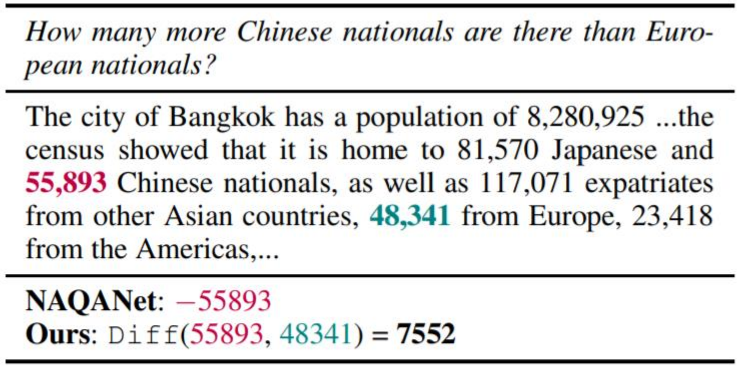
Daniel Andor, Luheng He, Kenton Lee, Emily Pitler
Google Research
{andor, luheng, kentonl, epitler}@google.com

EMNLP-IJCNLP 2019

Giving BERT a Calculator : Finding Operations and Arguments with Reading Comprehension [EMNLP 2019]

- It is unclear how best to generalize RC models to **abstractive numerical answers**
  - DROP dataset demonstrates that as long as there is **quantitative reasoning** involved, there are plenty of relatively straight forward questions that current extractive QA systems find difficult to answer.
  - Recent work has shown that SOTA neural models struggle with numerical operations and quantitative reasoning when trained in end-to-end manner.

  - The authors augment the model with a **predefined set of executable programs** which encompass simple arithmetic as well as extraction.
  - The program additionally provides a thin layer of interpretability that mirrors some of the reasoning required for the answer.



How many more Chinese nationals are there than European nationals?

The city of Bangkok has a population of 8,280,925 ...the census showed that it is home to 81,570 Japanese and **55,893** Chinese nationals, as well as 117,071 expatriates from other Asian countries, **48,341** from Europe, 23,418 from the Americas,...

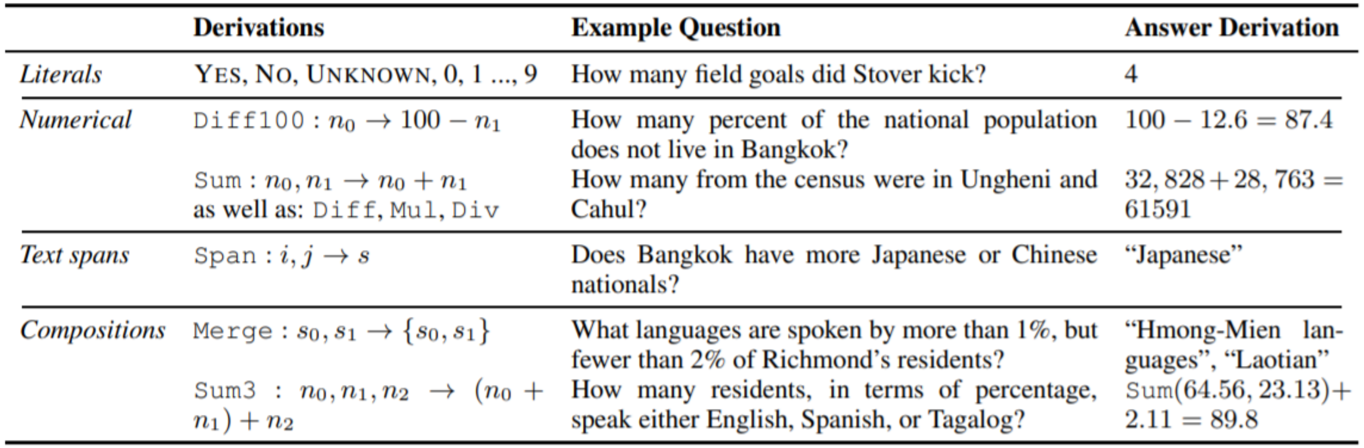
NAQANet: −55893
Ours: `Diff`(55893, 48341) = **7552**

Table 1: Example from the DROP development set. The correct answer is not explicitly stated in the passage and instead must be computed. The NAQANet model[2](Dua et al., 2019) predicts a negative number of people, whereas our model predicts that an operation `Diff` should be taken and identifies the two arguments.

Giving BERT a Calculator : Finding Operations and Arguments with Reading Comprehension [EMNLP 2019]

- Define the set of possible derivations $\mathcal{D}$
  - Could extend by **recursively searching** for compositions with deep derivations.
  - In this research, the system is guided by what is required in the DROP data and simply inference by heavily restricting multi-step compositions.

- Representations and Scoring
  - Literals : $\rho(d) = w_d^{\intercal} MLP_{lit}(h_{CLS})$ / Numerical operations : $\rho(d) = w_{op}^{\intercal} MLP_{binary}(h_i,\ h_j,\ h_i \odot h_j)$
  - Text spans : $\rho(d) = w_{span}^{\intercal} MLP_{span}(h_i,\ h_j)$ / Sum3 : $\rho(d) = w_{Sum3}^{\intercal} MLP_{Sum3}(h_{d0},\ h_k)$ / Merge : $\rho(d) = w_{Merge}^{\intercal} MLP_{Merge}(h_{d0},\ h_{d1},\ h_{d0} \odot h_{d1})$
  - Different with baseline(NAQANet) is that the authors use BERT as base encoder, and all derivations are all modeled, **allowing generalization** to new operations.



	Derivations	**Example Question**	**Answer Derivation**
Literals	YES, NO, UNKNOWN, 0, 1 ..., 9	How many field goals did Stover kick?	4
Numerical	`Diff100` : $n_0 \rightarrow 100 - n_1$	How many percent of the national population does not live in Bangkok?	$100 - 12.6 = 87.4$
	`Sum` : $n_0, n_1 \rightarrow n_0 + n_1$ as well as: `Diff`, `Mul`, `Div`	How many from the census were in Ungheni and Cahul?	$32,828 + 28,763 = 61591$
Text spans	`Span` : $i, j \rightarrow s$	Does Bangkok have more Japanese or Chinese nationals?	"Japanese"
Compositions	`Merge` : $s_0, s_1 \rightarrow \{s_0, s_1\}$	What languages are spoken by more than 1%, but fewer than 2% of Richmond's residents?	"Hmong-Mien languages", "Laotian"
	`Sum3` : $n_0, n_1, n_2 \rightarrow (n_0 + n_1) + n_2$	How many residents, in terms of percentage, speak either English, Spanish, or Tagalog?	`Sum`$(64.56, 23.13) + 2.11 = 89.8$

Table 2: Operations supported by the model. s, n refer to arguments of type *span* and *number*, respectively. i, j are the start and end indices of span s. The omitted definitions of `Diff`, `Mul`, and `Div` are analogous to `Sum`.

Giving BERT a Calculator : Finding Operations and Arguments with Reading Comprehension [EMNLP 2019]

- Training
  - **Marginalize out** all derivations d^* that lead to the answer.
  - If no derivation lead to the golden answer (where $\mathcal{D}^*$ is empty), skip the example.

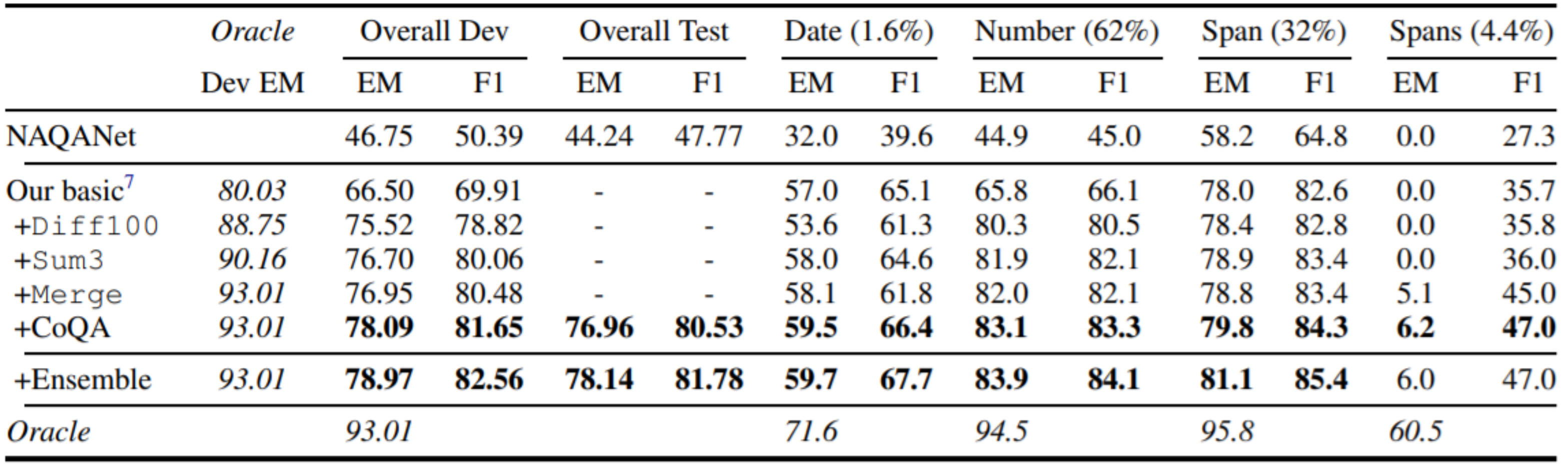
$$\mathcal{J}(P, Q, \mathcal{D}^*) = -\log \sum_{d^* \in \mathcal{D}^*} P(d^* \mid P, Q)$$

$$P(d \mid P, Q) = \frac{\exp \rho(d, P, Q)}{\sum_{d'} \exp \rho(d', P, Q)}$$

  - The **possible set** of $Merge$ is **quadratic** in the number $|S|$ of possible spans.
  - To do training and inference efficiently, only keep the top 128 $Span$ and Sum results when computing $Merge$ and $Sum3$.
  - During training, the pruned arguments had recall of 80~90% after 1 epoch and plateaued at 95~98%

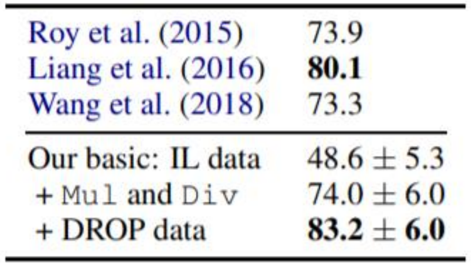
Giving BERT a Calculator : Finding Operations and Arguments with Reading Comprehension [EMNLP 2019]

- Experiments and Results
  - The authors used additional CoQA data for training.

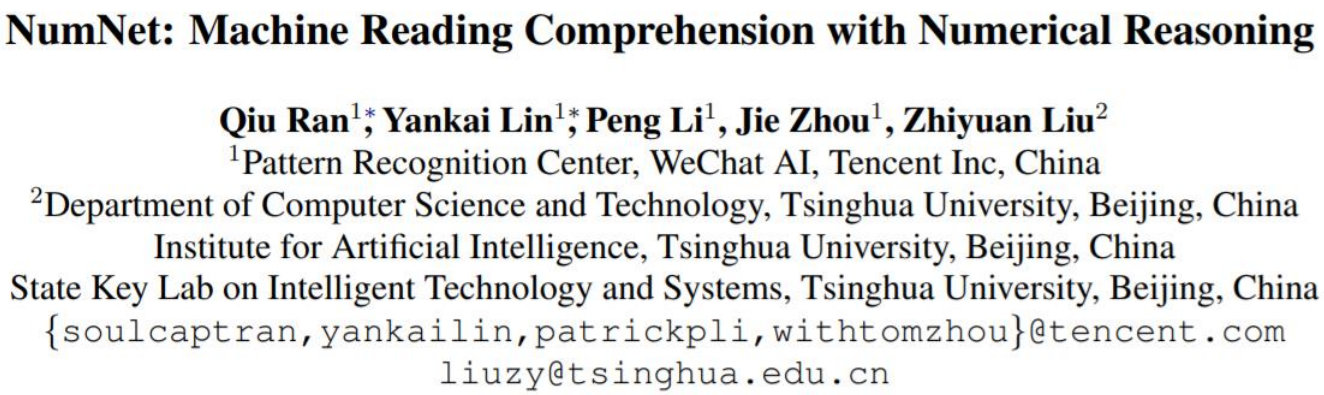


	Oracle	Overall Dev		Overall Test		Date (1.6%)		Number (62%)		Span (32%)		Spans (4.4%)	
	Dev EM	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
NAQANet		46.75	50.39	44.24	47.77	32.0	39.6	44.9	45.0	58.2	64.8	0.0	27.3
Our basic[7]	*80.03*	66.50	69.91	-	-	57.0	65.1	65.8	66.1	78.0	82.6	0.0	35.7
+`Diff100`	*88.75*	75.52	78.82	-	-	53.6	61.3	80.3	80.5	78.4	82.8	0.0	35.8
+`Sum3`	*90.16*	76.70	80.06	-	-	58.0	64.6	81.9	82.1	78.9	83.4	0.0	36.0
+`Merge`	*93.01*	76.95	80.48	-	-	58.1	61.8	82.0	82.1	78.8	83.4	5.1	45.0
+CoQA	*93.01*	**78.09**	**81.65**	**76.96**	**80.53**	**59.5**	**66.4**	**83.1**	**83.3**	**79.8**	**84.3**	**6.2**	**47.0**
+Ensemble	*93.01*	**78.97**	**82.56**	**78.14**	**81.78**	**59.7**	**67.7**	**83.9**	**84.1**	**81.1**	**85.4**	6.0	47.0
Oracle		*93.01*				*71.6*		*94.5*		*95.8*		*60.5*	

  - The authors also performed few-shot learning experiments on the Illinois dataset of math problems. (Requires multiplication, division which is not present in DROP)



Roy et al. (2015)	73.9
Liang et al. (2016)	**80.1**
Wang et al. (2018)	73.3
Our basic: IL data	48.6 ± 5.3
+ `Mul` and `Div`	74.0 ± 6.0
+ DROP data	**83.2 ± 6.0**



NumNet: Machine Reading Comprehension with Numerical Reasoning

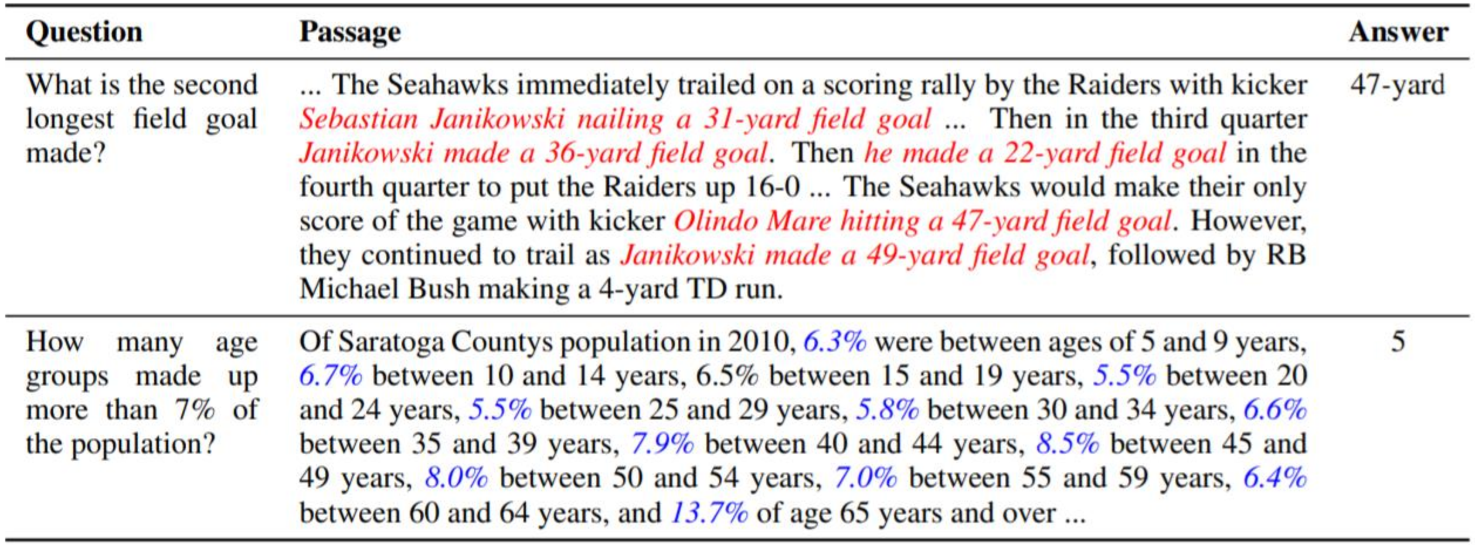
Qiu Ran[1]*, **Yankai Lin**[1]*, **Peng Li**[1], **Jie Zhou**[1], **Zhiyuan Liu**[2]
[1]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
{soulcaptran,yankailin,patrickpli,withtomzhou}@tencent.com
liuzy@tsinghua.edu.cn

EMNLP-IJCNLP 2019

NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- **Numerical reasoning** is a critical skill in human's RC, which has not been well considered in current MRC systems
  - NAQANet makes a pioneering attempt to answer numerical questions but still does not explicitly consider numerical reasoning.
  - To best of the author's knowledge, this work is the first one that **explicitly incorporates numerical reasoning** into the MRC system.

  - A key problem to answer questions requiring numerical reasoning is how to perform **numerical comparison** in MRC systems, which is crucial for two common types of questions in the DROP dataset.
  - 1) **Numerical Comparison** : Answers of the questions could be directly obtained via performing numerical comparison (e.g., Sorting)
  - 2) **Numerical Condition** : Answers of the questions cannot be directly obtained through simple numerical comparison, but often require numerical comparison for understanding the text.



Question	**Passage**	**Answer**
What is the second longest field goal made?	... The Seahawks immediately trailed on a scoring rally by the Raiders with kicker *Sebastian Janikowski nailing a 31-yard field goal* ... Then in the third quarter *Janikowski made a 36-yard field goal*. Then *he made a 22-yard field goal* in the fourth quarter to put the Raiders up 16-0 ... The Seahawks would make their only score of the game with kicker *Olindo Mare hitting a 47-yard field goal*. However, they continued to trail as *Janikowski made a 49-yard field goal*, followed by RB Michael Bush making a 4-yard TD run.	47-yard
How many age groups made up more than 7% of the population?	Of Saratoga Countys population in 2010, *6.3%* were between ages of 5 and 9 years, *6.7%* between 10 and 14 years, 6.5% between 15 and 19 years, *5.5%* between 20 and 24 years, *5.5%* between 25 and 29 years, *5.8%* between 30 and 34 years, *6.6%* between 35 and 39 years, *7.9%* between 40 and 44 years, *8.5%* between 45 and 49 years, *8.0%* between 50 and 54 years, *7.0%* between 55 and 59 years, *6.4%* between 60 and 64 years, and *13.7%* of age 65 years and over ...	5

NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Model Architecture
  - The model is composed of encoding module, reasoning module and prediction module.
  - The major contribution of this work is the reasoning module, which leverages NumGNN between the encoding module and prediction module to explicitly **consider the numerical comparison information** and perform numerical reasoning.

  - **(Encoding Module)** W.L.O.G, the authors use encoding components of QANet and NAQANet to encode the question and passage into vector representations.
  - While the first encoding module is composed of "Convolution – Self-Attention – Feed-forward Layers", the second encoding module is a passage-question attention layer.

$$\boldsymbol{Q} = \texttt{QANet-Emb-Enc}(Q), \quad (1) \qquad \bar{\boldsymbol{Q}} = \texttt{QANet-Att}(\boldsymbol{P}, \boldsymbol{Q}), \quad (3)$$

$$\boldsymbol{P} = \texttt{QANet-Emb-Enc}(P), \quad (2) \qquad \bar{\boldsymbol{P}} = \texttt{QANet-Att}(\boldsymbol{Q}, \boldsymbol{P}), \quad (4)$$

  - **(Reasoning Module; NumGNN)** Building a **heterogeneous directed graph** whose nodes are corresponding to numbers in the question and passage, the edges are used to encode numerical relationships among the numbers.

$$\boldsymbol{M}^Q = \texttt{QANet-Mod-Enc}(\boldsymbol{W}^M \bar{\boldsymbol{Q}}), \quad (5)$$

$$\boldsymbol{M}^P = \texttt{QANet-Mod-Enc}(\boldsymbol{W}^M \bar{\boldsymbol{P}}), \quad (6)$$

$$\boldsymbol{U} = \texttt{Reasoning}(\mathcal{G}; \boldsymbol{M}^Q, \boldsymbol{M}^P), \quad (7)$$

  - As U only contains the representations of numbers, to tackle span-style answers containing non-numerical words, the authors concatenate U with M^P to produce **numerically-aware passage representation** M_0.

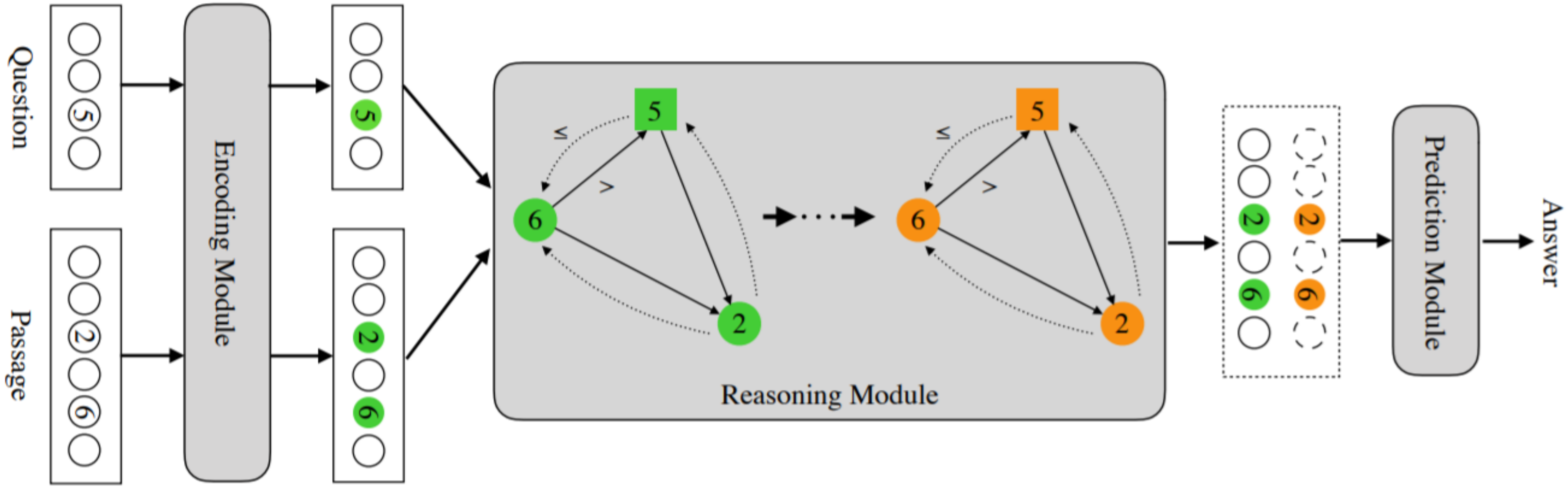
$$\boldsymbol{M}^{\text{num}}[i] = \begin{cases} \boldsymbol{U}[I(i)] & \text{if } w_i^p \text{ is a number} \\ \boldsymbol{0} \end{cases},$$

$$\boldsymbol{M}_0' = \boldsymbol{W}_0[\boldsymbol{M}^P; \boldsymbol{M}^{\text{num}}] + \boldsymbol{b}_0, \quad (8)$$

$$\boldsymbol{M}_0 = \texttt{QANet-Mod-Enc}(\boldsymbol{M}_0'), \quad (9)$$

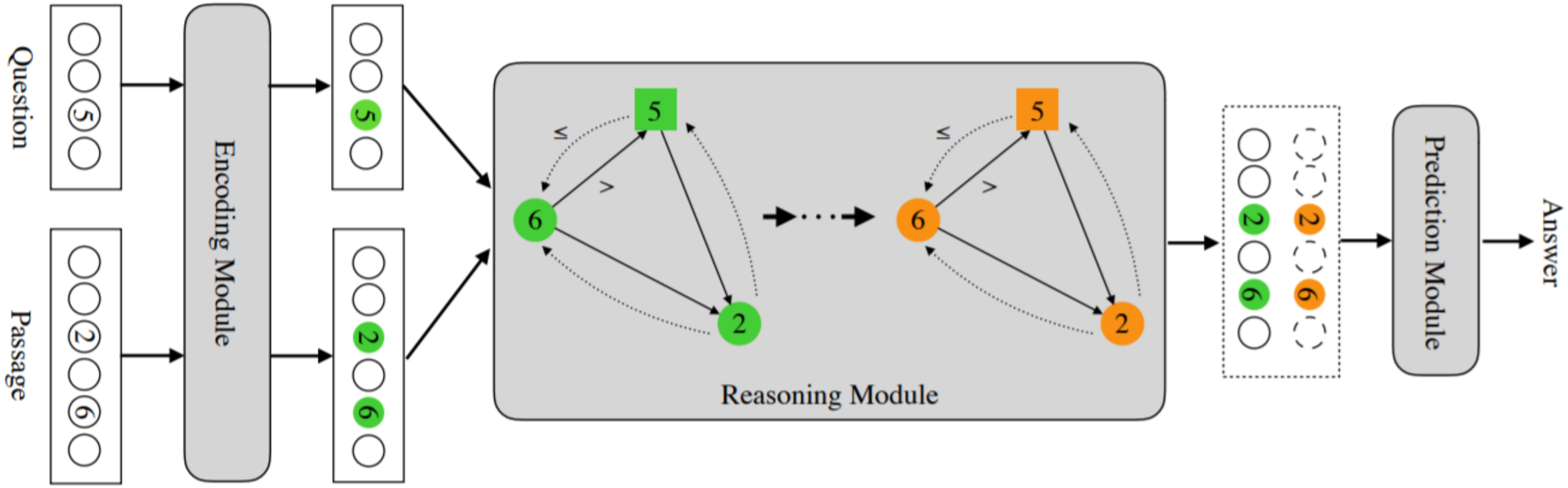
NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Model Architecture
  - **(Prediction Module)** Like in NAQANet, the authors divide the answers into four types and use unique output layer to calculate the conditional answer probability for each type.
  - Answer Types : 1) Passage span, 2) Question span, 3) Count, 4) Arithmetic expression (Answer is result of an arithmetic expression)
  - Meanwhile, an extra output layer is also used to predict the probability of the type of each answer.
  - At training time, the final answer probability is defined as the joint probability over all feasible answer types, where at test time, the model first selects the most probable answer type greedily and then predicts the best answer accordingly.



NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Numerically-aware Graph Construction (NumGNN)
    - **Two set of edges** are considered in this work : Greater Relation Edge, Lower or Equal Relation Edge
    - Theoretically, these two types are **complement** to each other.
    - However, as a number may occur several times and represent different facts in a document, we add a distinct node for each occurrence in the graph to prevent **potential ambiguity**.
    - Therefore, it is more reasonable to use both types in order to encode the equal information among nodes.



NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Numerical Reasoning
  - The initial representation is the encoded representation from the Encoding Module.
  - Given a graph, the authors use a GNN to perform reasoning in three steps.
  - **(Step1 : Node Relatedness Measure)** Since only a few numbers are relevant for answering a question generally, **compute the weight for each node to by-pass** irrelevant numbers in reasoning.

$$\alpha_i = \mathrm{sigmoid}(\boldsymbol{W}_v \boldsymbol{v}[i] + b_v), \qquad (10)$$

  - **(Step2 : Message Propagation)** As the role a number plays in reasoning is not only decided by itself, but also **related to the context**, propagate messages from each node to its neighbors to help to perform reasoning.
  - As numbers in question and passage may play different roles in reasoning and edges corresponding to different numerical relations should be distinguished, the authors use relation-specific transform matrices in the message propagation.
  - **(Node types)** 1) Both from question; 2) Both from passage; 3) From question and the passage respectively ; 4) From passage and question respectively.

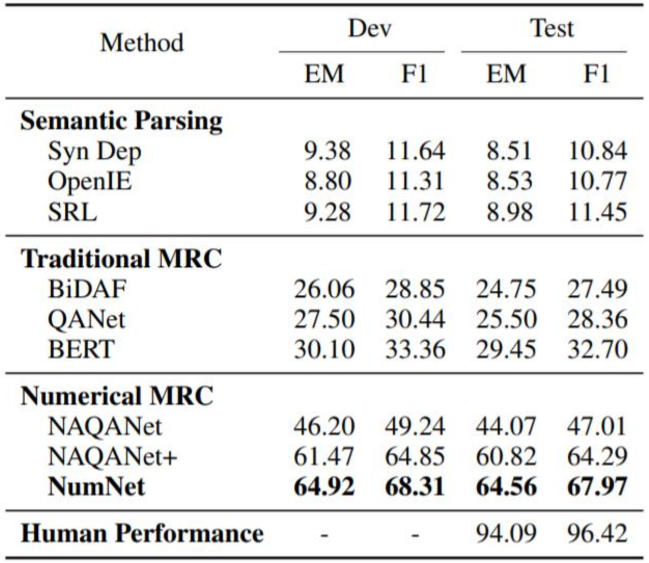
$$\widetilde{\boldsymbol{v}}'_i = \frac{1}{|\mathcal{N}_i|}\left(\sum_{j\in\mathcal{N}_i} \alpha_j \boldsymbol{W}^{r_{ji}} \boldsymbol{v}[j]\right), \qquad (11)$$

  - **(Step3 : Node Representation Update)** As the message representation obtained in the previous step only contains information from the neighbors, it needs to be fused with node representation to **combine with the information carried by the node itself**.

$$\boldsymbol{v}'_i = \mathrm{ReLU}(\boldsymbol{W}_f \boldsymbol{v}_i + \widetilde{\boldsymbol{v}}'_i + \boldsymbol{b}_f), \qquad (12)$$

NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Experiments and Results
  - Evaluate on DROP dataset which require numerical reasoning such as addition, counting, or sorting over numbers in passage.
  - The reasoning step K in NumGNN is set to 3.

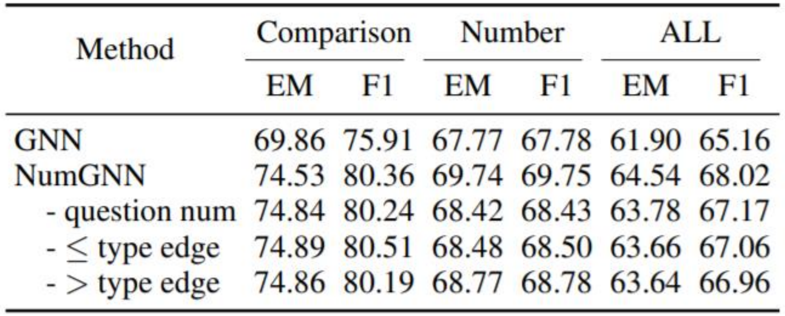


Method	Dev		Test	
	EM	F1	EM	F1
Semantic Parsing				
Syn Dep	9.38	11.64	8.51	10.84
OpenIE	8.80	11.31	8.53	10.77
SRL	9.28	11.72	8.98	11.45
Traditional MRC				
BiDAF	26.06	28.85	24.75	27.49
QANet	27.50	30.44	25.50	28.36
BERT	30.10	33.36	29.45	32.70
Numerical MRC				
NAQANet	46.20	49.24	44.07	47.01
NAQANet+	61.47	64.85	60.82	64.29
NumNet	**64.92**	**68.31**	**64.56**	**67.97**
Human Performance	-	-	94.09	96.42

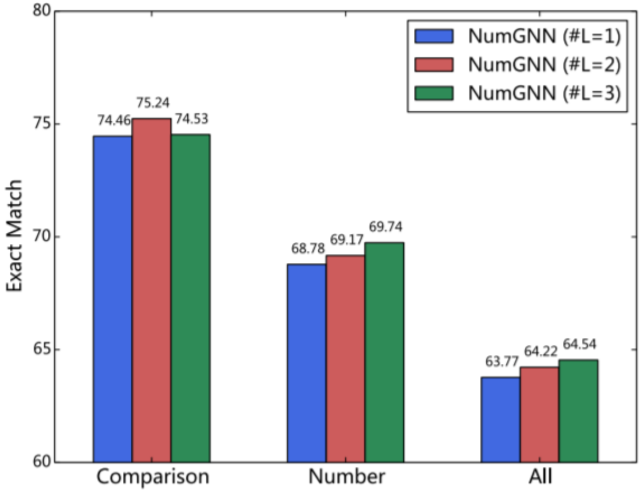
  - NumNet model achieves better results compared to semantic parsing-based models, traditional MRC models and even numerical MRC models(NAQANet).
  - The authors implemented an advanced version of NAQANet, which is NAQANet+ that considers real numbers, rich arithmetic expression, data augmentation, etc.

NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Ablation Studies
  - If replacing numerically-aware graph with a fully connected graph, the model fallbacks to a traditional GNN.
  - As shown in the results, the proposed NumGNN leads to statistically significant improvements compared to traditional GNN.
  - 2-Layer version of NumNet achieves best performance, because the questions in DROP require at most 2-step reasoning.
  - Although performance improves as number of layer increases, further investigation shows that performance gain is not stable when K>=4 due to the intrinsic over smoothing problem of GNNs.

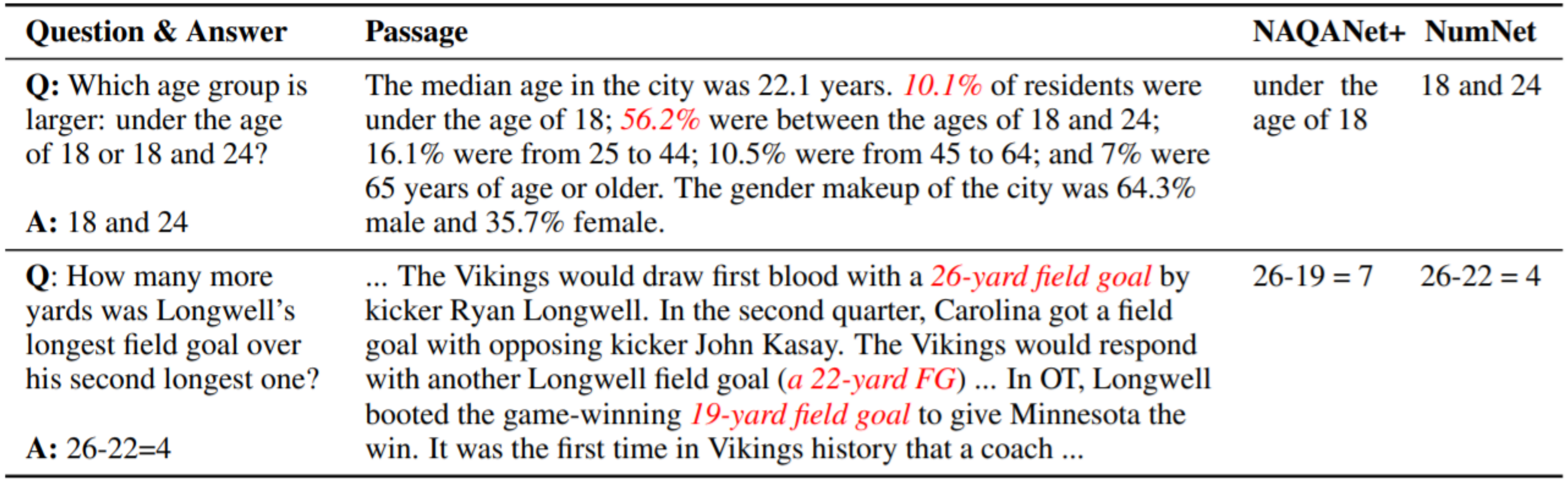


Method	Comparison		Number		ALL	
	EM	F1	EM	F1	EM	F1
GNN	69.86	75.91	67.77	67.78	61.90	65.16
NumGNN	74.53	80.36	69.74	69.75	64.54	68.02
- question num	74.84	80.24	68.42	68.43	63.78	67.17
- $\leq$ type edge	74.89	80.51	68.48	68.50	63.66	67.06
- $>$ type edge	74.86	80.19	68.77	68.78	63.64	66.96



NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Case Studies

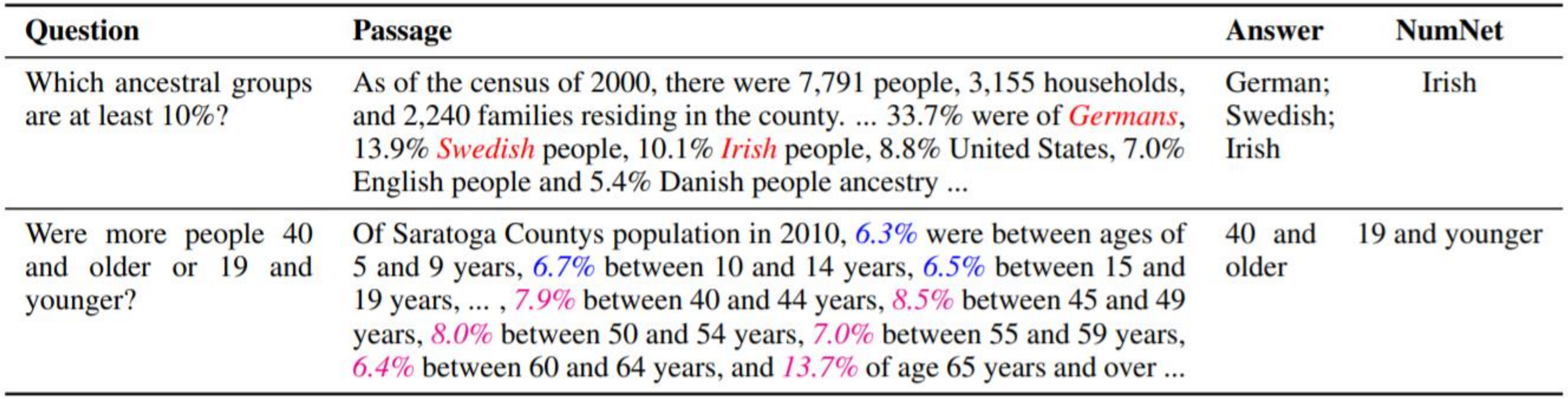


Question & Answer	**Passage**	**NAQANet+**	**NumNet**
Q: Which age group is larger: under the age of 18 or 18 and 24? **A:** 18 and 24	The median age in the city was 22.1 years. *10.1%* of residents were under the age of 18; *56.2%* were between the ages of 18 and 24; 16.1% were from 25 to 44; 10.5% were from 45 to 64; and 7% were 65 years of age or older. The gender makeup of the city was 64.3% male and 35.7% female.	under the age of 18	18 and 24
Q: How many more yards was Longwell's longest field goal over his second longest one? **A:** 26-22=4	... The Vikings would draw first blood with a *26-yard field goal* by kicker Ryan Longwell. In the second quarter, Carolina got a field goal with opposing kicker John Kasay. The Vikings would respond with another Longwell field goal (*a 22-yard FG*) ... In OT, Longwell booted the game-winning *19-yard field goal* to give Minnesota the win. It was the first time in Vikings history that a coach ...	26-19 = 7	26-22 = 4

Table 4: Cases from the DROP dataset. We demonstrate the predictions of NAQANet+ and our NumNet model. Note that the two models only output the arithmetic expressions but we also provide their results for clarity.

NumNet : Machine Reading Comprehension with Numerical Reasoning [EMNLP 2019]

- Case Studies (Error Analysis)
  - Since the numerically-aware graph is pre-defined, NumNet is not applicable to the case where an intermediate number has to be derived.

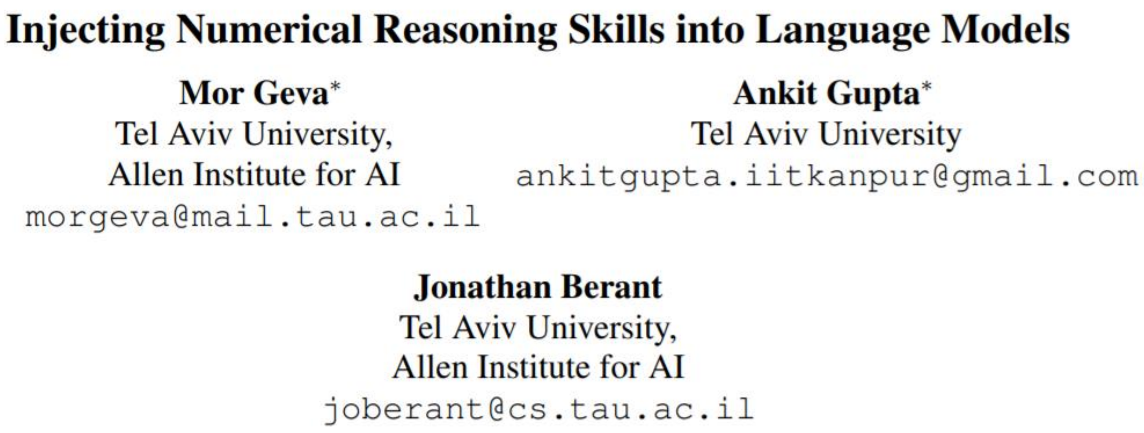


Question	Passage	Answer	NumNet

Which ancestral groups are at least 10%?	As of the census of 2000, there were 7,791 people, 3,155 households, and 2,240 families residing in the county. ... 33.7% were of *Germans*, 13.9% *Swedish* people, 10.1% *Irish* people, 8.8% United States, 7.0% English people and 5.4% Danish people ancestry ...	German;
		Swedish;
		Irish
			Irish

Were more people 40 and older or 19 and younger?	Of Saratoga Countys population in 2010, *6.3%* were between ages of 5 and 9 years, *6.7%* between 10 and 14 years, *6.5%* between 15 and 19 years, ... , *7.9%* between 40 and 44 years, *8.5%* between 45 and 49 years, *8.0%* between 50 and 54 years, *7.0%* between 55 and 59 years, *6.4%* between 60 and 64 years, and *13.7%* of age 65 years and over ...	40 and older	19 and younger

Table 5: Typical error examples. Row 1: the answer is multiple nonadjacent spans; Row 2: Intermediate numbers are involved in reasoning.



Injecting Numerical Reasoning Skills into Language Models

Mor Geva[*]
Tel Aviv University,
Allen Institute for AI
morgeva@mail.tau.ac.il

Ankit Gupta[*]
Tel Aviv University
ankitgupta.iitkanpur@gmail.com

Jonathan Berant
Tel Aviv University,
Allen Institute for AI
joberant@cs.tau.ac.il

ACL 2020

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Existing Models for numerical reasoning have used specialized architectures with **limited flexibility**.
  - These models use modules that are designed for counting (but only until '9') and for addition and subtraction (but of 2~3 numbers only).
  - Such models perform well on existing datasets, but **do not generalize to unsupported computations**.
  - Current models marginalize at training time over all numerical expressions that evaluate to the correct answer,
    but since the number of such expressions **grows exponentially**, scaling these approaches to arbitrary computations entails using non-differentiable operations.
  - Delegating numerical computations to an external symbolic calculator leads to **modeling challenges** (e.g., "How threw 45 total yards for touchdowns?")
  - In this work, the authors propose that reasoning skills, such as numerical reasoning, are amenable to **automatic data generation**.
  - One can inject that skill directly into the model by adding additional pre-training steps, allowing the model to learn the skill in an **end-to-end fashion**.
- The authors add to a large PTLM **two pre-training steps** over automatically-generated synthetic data.
  - Firstly, generating numerical data of the form "3+4+11=18" and training the model teaches it to compute the value of numbers from their tokens and to **perform numerical operations**.
  - Secondly, generating question-passage pairs that require numerical reasoning using a compact grammar endows the model with the ability to understand computations expressed in **pseudo-natural language**.
  - In both pre-training steps, the model(GEN-BERT) generates output numbers token-by-token,
    so an answer can either be extracted from the input **with an encoder** or generated **from a decoder**.
  - Pretraining is done in a **multi-task setup** with a standard LM objective, in order to avoid "catastrophic forgetting".

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Model Architecture
  - A BERT-based generative model(Encoder-Decoder architecture) that performs numerical computations internally, termed GENBERT.
  - To enjoy BERT's representations at decoding time, we tie the weights of the decoder and the encoder. (Initialize BERT's weights to both encoder, decoder)

  - Since the encoder and decoder weights are tied, we make them learn distinct representations by adding a $FFN\ FF_{enc}$ that transforms encoder's representation.

$$\mathbf{H_{enc}} = \texttt{layer-norm}(\texttt{gelu}(W \cdot \mathbf{L_{enc}})),$$

  - To further distinguish the encoder and decoder, the authors use distinct start and end tokens for input and output sequences ([SOS], [EOS]).
  - The output tokens pass through the decoder and FF_{dec} to obtain H_{dec}.

  - The decoder outputs the probability $p_{dec}(a_{i+1}|a_0, a_1, \dots, a_i, c, q)$ and this value is used to acquire the probability of an answer.

$$p_{\text{dec}}(\langle \mathbf{a} \rangle \mid \mathbf{c}, \mathbf{q}) = \prod_{i=0}^{m} p_{\text{dec}}(a_{i+1} \mid a_0, ..a_i, \mathbf{c}, \mathbf{q}).$$

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Model Architecture

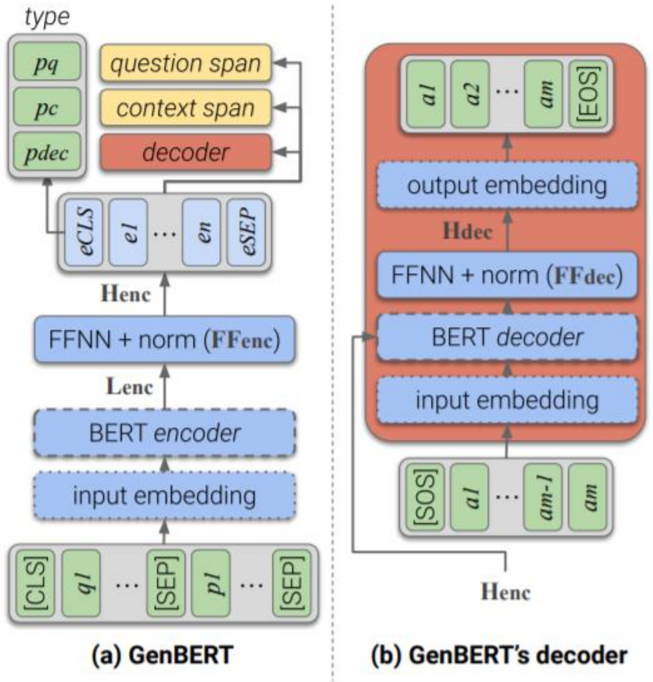


Figure 2: GENBERT's network architecture: (a) a high-level overview of the network, including a generative head (red), two span-extraction heads (yellow), and an answer type head. (b) a closer overview of GENBERT's generative head.

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

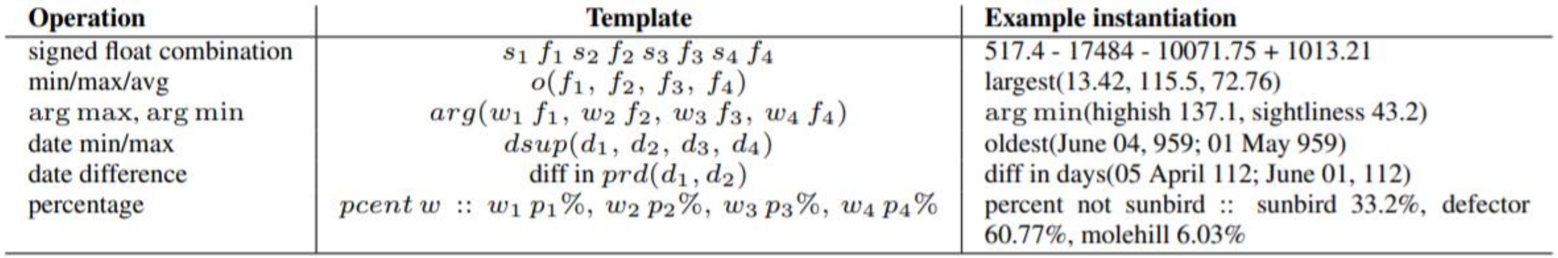
- Model Architecture
  - To improve pre-training on numeric data, two additional modifications are made.
  - **Digit Tokenization**
  - Conventional word-piece tokenization treats numbers no differently than any other token.
  - However, computing the value of numbers should be simpler when using digits directly.
  - For example, a wordpiece $\#\#d_1 \ldots d_k$ is further splitted into $\#\#d_1, \#\#d_2, \ldots, \#\#d_k$.
  - **Random Shift**
  - With short inputs such as "1086.1 – 2.54 + 343.8", the model can potentially over-fit and learn to perform numerical reasoning only when numbers are at the beginning of an input.
  - To prevent this, when the input length is shorter than 512, the authors shift all position IDs by an random integer in (0,1,…,512-|L|)

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Training
  - For each span (i,j), a span extraction head outputs its probability of being the answer.
  - GENBERT obtains 46.1 EM and 49.3 F1, roughly 20 points lower than prior models, and conclude that it is hard to obtain numerical reasoning skills with only DROP.

$$-\log\left(p_{\mathbf{dec}}\cdot p_{\text{dec}}(\langle\mathbf{a}\rangle)+\sum_{\mathbf{h}\in\mathbf{q},\mathbf{c}} p_{\mathbf{h}}\cdot\sum_{(i,j)\in S} p_h(i,j)\right),$$

- Generating Numerical Data
  - The first dataset focuses on learning numerical values expressed by tokens and computing numerical operations. (Does not involve textual content)
  - Each template consists of an expression to evaluate and its solution.



Operation	Template	Example instantiation
signed float combination	$s_1\ f_1\ s_2\ f_2\ s_3\ f_3\ s_4\ f_4$	517.4 - 17484 - 10071.75 + 1013.21
min/max/avg	$o(f_1,\ f_2,\ f_3,\ f_4)$	largest(13.42, 115.5, 72.76)
arg max, arg min	$arg(w_1\ f_1,\ w_2\ f_2,\ w_3\ f_3,\ w_4\ f_4)$	arg min(highish 137.1, sightliness 43.2)
date min/max	$dsup(d_1,\ d_2,\ d_3,\ d_4)$	oldest(June 04, 959; 01 May 959)
date difference	diff in $prd(d_1, d_2)$	diff in days(05 April 112; June 01, 112)
percentage	$pcent\ w\ ::\ w_1\ p_1\%,\ w_2\ p_2\%,\ w_3\ p_3\%,\ w_4\ p_4\%$	percent not sunbird :: sunbird 33.2%, defector 60.77%, molehill 6.03%

Table 2: Templates for generating synthetic numerical examples and the numerical operations required to answer them. **Domains** (defined in App. A.1): $s_i \in \{-,+\}$, $f_i \in \mathbb{R}^+$, $o \in \mathcal{O}$: superlative words like *"longest"*, $arg \in \{\text{arg min}, \text{arg max}\}$, $w_i \in \mathcal{W}$: words from NTLK Words Corpus, $d_i \in \mathcal{D}$: dates until Sep 2019, $dsup \in \mathcal{DSUP}$: superlative words like *"latest"*, $prd \in \{$*"days"*, *"months"*, *"years"*$\}$, $p_i \in (0, 100)$, $pcent \in \{$*"percent"*, *"percent not"*$\}$.

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Generating Textual Data
  - To tackle NRoT(Numerical Reasoning on Text), a model needs to **comprehend how numerical operations are expressed in text** that refers to events, entities and quantities.
  - While **text generation** is hard in the general case, we are specifically interested in text that focuses on number manipulations.
  - Use the framework of *Hosseini et al.(2014)*, who proposed to model math word problems with a simple structure.
  - Going over sentences from the corpus, use procedure to abstract its tokens into categories, and count for each **extracted template** its frequency in data.
  - Using the top-12 extracted template, **generate passages** with a small vocabulary, and **generate questions** with 13 question templates.

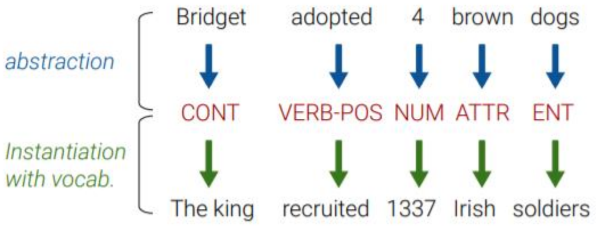
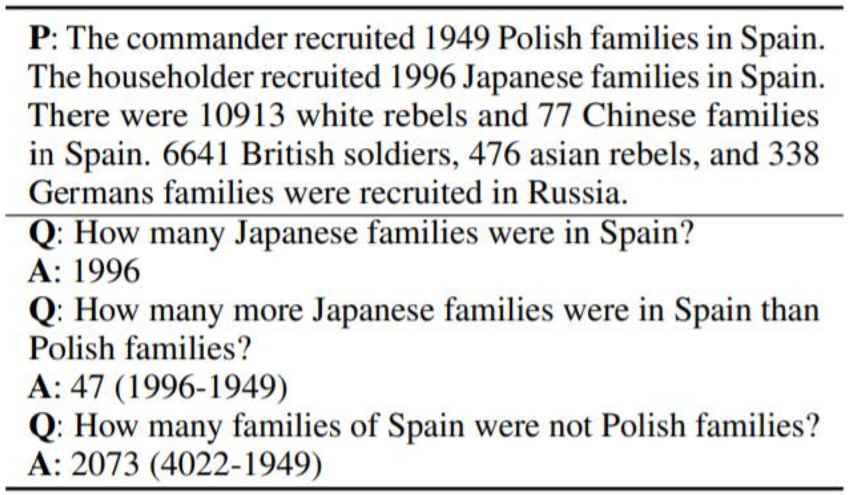


Figure 3: Template extraction and instantiation. A template (in red) is extracted from a MWP sentence, using categories for containers, entities, verbs, attributes and numbers, according to Hosseini et al. (2014). For generation, the categories are instantiated with a domain-specific vocabulary.



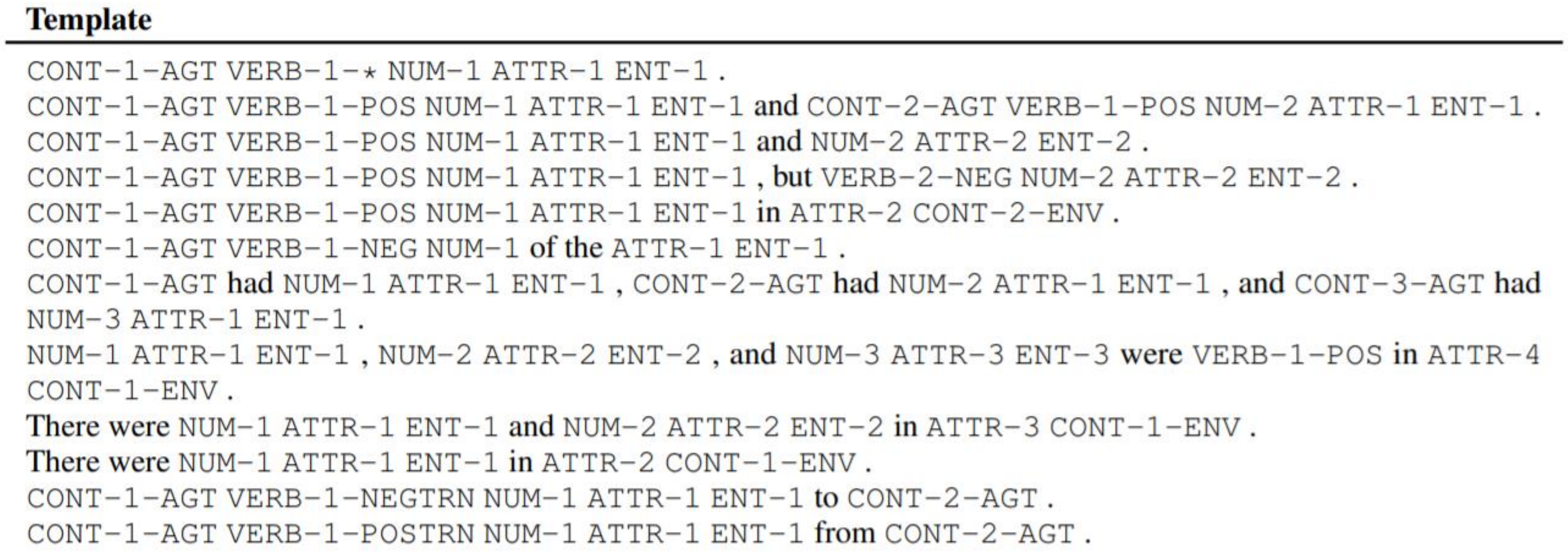
P: The commander recruited 1949 Polish families in Spain. The householder recruited 1996 Japanese families in Spain. There were 10913 white rebels and 77 Chinese families in Spain. 6641 British soldiers, 476 asian rebels, and 338 Germans families were recruited in Russia.

Q: How many Japanese families were in Spain?
A: 1996
Q: How many more Japanese families were in Spain than Polish families?
A: 47 (1996-1949)
Q: How many families of Spain were not Polish families?
A: 2073 (4022-1949)

Table 3: An example synthetic passage (P) and questions. Questions (Q) were generated from templates and answers (A) were calculated based on the world state.

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Generating Textual Data

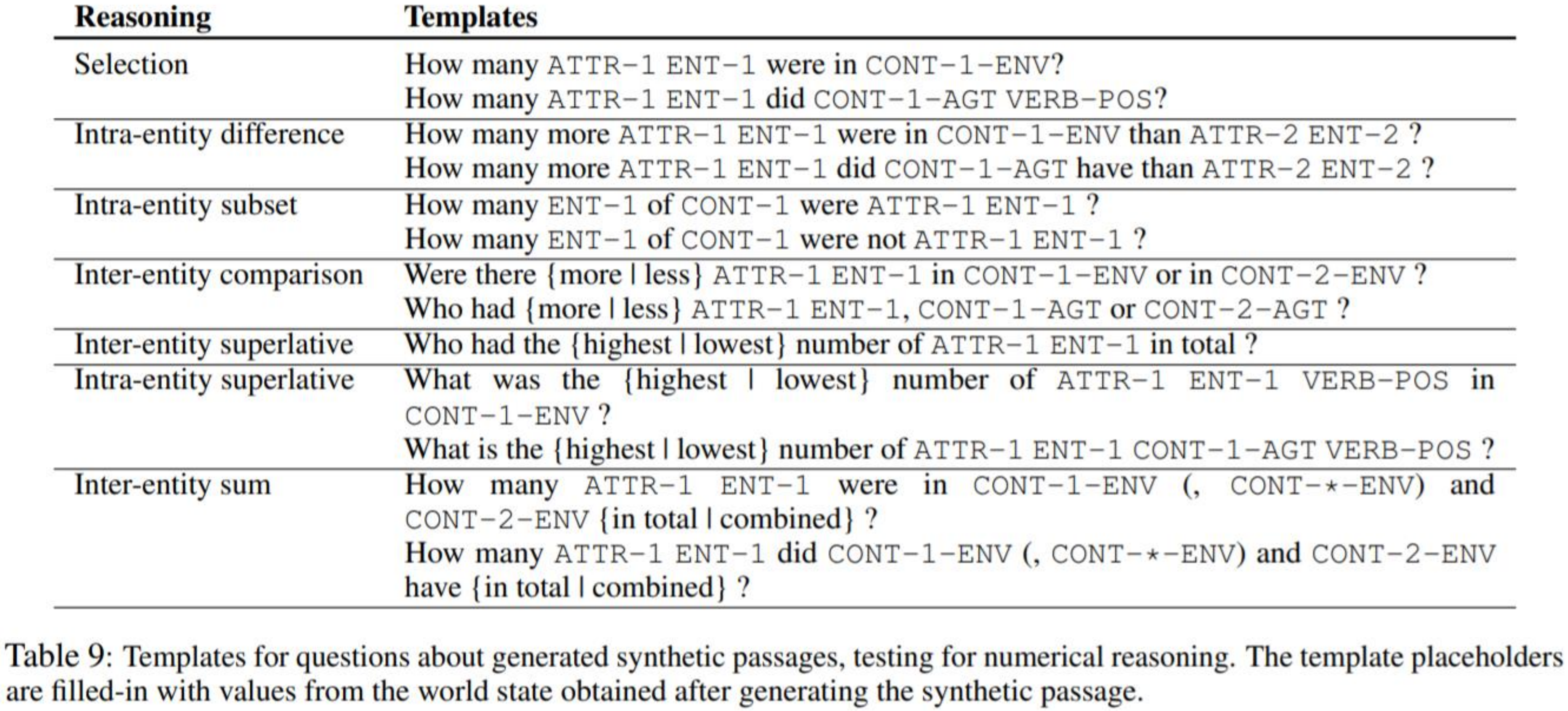


Template
CONT-1-AGT VERB-1-* NUM-1 ATTR-1 ENT-1 .
CONT-1-AGT VERB-1-POS NUM-1 ATTR-1 ENT-1 and CONT-2-AGT VERB-1-POS NUM-2 ATTR-1 ENT-1 .
CONT-1-AGT VERB-1-POS NUM-1 ATTR-1 ENT-1 and NUM-2 ATTR-2 ENT-2 .
CONT-1-AGT VERB-1-POS NUM-1 ATTR-1 ENT-1 , but VERB-2-NEG NUM-2 ATTR-2 ENT-2 .
CONT-1-AGT VERB-1-POS NUM-1 ATTR-1 ENT-1 in ATTR-2 CONT-2-ENV .
CONT-1-AGT VERB-1-NEG NUM-1 of the ATTR-1 ENT-1 .
CONT-1-AGT had NUM-1 ATTR-1 ENT-1 , CONT-2-AGT had NUM-2 ATTR-1 ENT-1 , and CONT-3-AGT had NUM-3 ATTR-1 ENT-1 .
NUM-1 ATTR-1 ENT-1 , NUM-2 ATTR-2 ENT-2 , and NUM-3 ATTR-3 ENT-3 were VERB-1-POS in ATTR-4 CONT-1-ENV .
There were NUM-1 ATTR-1 ENT-1 and NUM-2 ATTR-2 ENT-2 in ATTR-3 CONT-1-ENV .
There were NUM-1 ATTR-1 ENT-1 in ATTR-2 CONT-1-ENV .
CONT-1-AGT VERB-1-NEGTRN NUM-1 ATTR-1 ENT-1 to CONT-2-AGT .
CONT-1-AGT VERB-1-POSTRN NUM-1 ATTR-1 ENT-1 from CONT-2-AGT .

Table 8: Sentence templates for synthetic textual examples.

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Generating Textual Data



Reasoning	Templates
Selection	How many ATTR−1 ENT−1 were in CONT−1−ENV? How many ATTR−1 ENT−1 did CONT−1−AGT VERB−POS?
Intra-entity difference	How many more ATTR−1 ENT−1 were in CONT−1−ENV than ATTR−2 ENT−2 ? How many more ATTR−1 ENT−1 did CONT−1−AGT have than ATTR−2 ENT−2 ?
Intra-entity subset	How many ENT−1 of CONT−1 were ATTR−1 ENT−1 ? How many ENT−1 of CONT−1 were not ATTR−1 ENT−1 ?
Inter-entity comparison	Were there {more | less} ATTR−1 ENT−1 in CONT−1−ENV or in CONT−2−ENV ? Who had {more | less} ATTR−1 ENT−1, CONT−1−AGT or CONT−2−AGT ?
Inter-entity superlative	Who had the {highest | lowest} number of ATTR−1 ENT−1 in total ?
Intra-entity superlative	What was the {highest | lowest} number of ATTR−1 ENT−1 VERB−POS in CONT−1−ENV ? What is the {highest | lowest} number of ATTR−1 ENT−1 CONT−1−AGT VERB−POS ?
Inter-entity sum	How many ATTR−1 ENT−1 were in CONT−1−ENV (, CONT−⋆−ENV) and CONT−2−ENV {in total | combined} ? How many ATTR−1 ENT−1 did CONT−1−ENV (, CONT−⋆−ENV) and CONT−2−ENV have {in total | combined} ?

Table 9: Templates for questions about generated synthetic passages, testing for numerical reasoning. The template placeholders are filled-in with values from the world state obtained after generating the synthetic passage.

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Training on Synthetic Data (Numerical Data, Textual Data)
  - For ND, generate 1M examples for training and 10K for validation
  - For TD, generate 2.5M examples for training and 10K for validation.
  - To ensure the model does not lose its Language Understanding abilities, employ a multi-task setup.
  - Concretely, while pre-training on ND and TD, also sample mini-batches from pretrain-corpus.

$$\mathbf{L}_{\text{mlm}}(\langle\mathbf{m}\rangle) = \text{mean}_{i\in\text{masked}} -\log(p(a_i \mid i, \langle\mathbf{m}\rangle)).$$

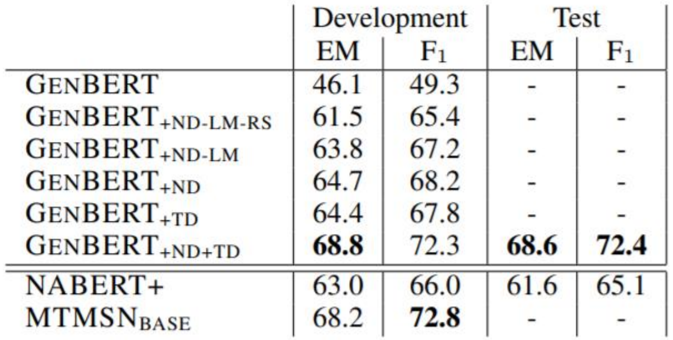
$$\mathbf{L}_{\text{model}}(X_{\text{ND}}) + \mathbf{L}_{\text{model}}(X_{\text{TD}}) + \lambda\cdot\mathbf{L}_{\text{mlm}}(X_{\text{MLM}}).$$

Injecting Numerical Reasoning Skills into Language Models [ACL 2020]

- Experiments and Results
  - GENBERT consistently achieves more than 96% accuracy in predicting correct solutions for ND and TD.
  - The authors conclude that a PTLM can learn the designed numerical reasoning skills from generated data through Pretraining.
  - Model already trained on ND converges faster on TD.

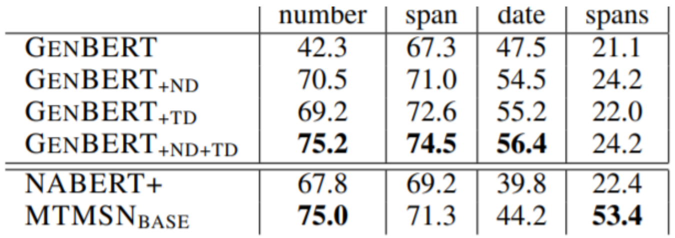
  - Test GENBERT guided by the following questions.
  - 1) Are the injected skills **robust** and **generalize to NRoT datasets** like DROP? => Finetune on DROP and further evaluate on MWP in a zero-shot setup.
  - 2) Are the new skills learned at the expense of the model's ability to **understand language**? => Evaluate GENBERT on SQuAD.
  - 3) Can the pre-trained weights be used with architectures other than BERT? => Use GENBERT encoder as a drop-in replacement for BERT on two other architectures.

  - Skills learn from ND and TD are complementary, and it is important to include MLM loss.



	Development		Test	
	EM	F$_1$	EM	F$_1$
GenBERT	46.1	49.3	-	-
GenBERT$_{\text{+ND-LM-RS}}$	61.5	65.4	-	-
GenBERT$_{\text{+ND-LM}}$	63.8	67.2	-	-
GenBERT$_{\text{+ND}}$	64.7	68.2	-	-
GenBERT$_{\text{+TD}}$	64.4	67.8	-	-
GenBERT$_{\text{+ND+TD}}$	**68.8**	72.3	**68.6**	**72.4**
NABERT+	63.0	66.0	61.6	65.1
MTMSN$_{\text{BASE}}$	68.2	**72.8**	-	-

Table 4: Performance of GenBERT and comparable models on the development and test sets of DROP.



	number	span	date	spans
GenBERT	42.3	67.3	47.5	21.1
GenBERT$_{\text{+ND}}$	70.5	71.0	54.5	24.2
GenBERT$_{\text{+TD}}$	69.2	72.6	55.2	22.0
GenBERT$_{\text{+ND+TD}}$	**75.2**	**74.5**	**56.4**	24.2
NABERT+	67.8	69.2	39.8	22.4
MTMSN$_{\text{BASE}}$	**75.0**	71.3	44.2	**53.4**

Table 5: F$_1$ scores on DROP development per answer type.

ANY QUESTIONS?