Project 3: DROP: reading comprehension

Wai Leuk Lo
BDT HKUST
wlloaa@connect.ust.hk

Abstract

We choose Project 3: DROP: reading comprehension as our project, introduce the task and its importance, describe our methodology to solve this task and finally analyze and discuss the results.

1 Introduction

The task of reading comprehension involves understanding a passage of text and answering questions based on it. This project studied a question-answer data set called DROP (Discrete Reasoning Over the text in the Paragraph) to further paragraph semantic understanding and symbolic reasoning. To perform well in this data set, the model must discern the semantics of a paragraph and perform one kind or a combination of discrete reasonings on them, where the reasonings include subtraction, comparison, selection, addition, count and sort, coreference resolution, other arithmetic, set of spans (a set of texts, each being a consecutive piece of words) and others.

DROP is developed by first extracting 7,000 Wikipedia passages containing a narrative sequence of events and many numbers, then collecting question-answer pairs with crowd-sourcing, notably with an adversarial annotation setting against the question-answer model BiDAF, and finally validating the quality of the data set using inter-annotator agreement. We use the summary table (Table 1) from the original paper (Dua et al., 2019) to illustrate the phenomena mentioned above.

Current models perform poorly on this dataset while humans obtain extreme high scores. This is due to the characteristics of the dataset:

- As an adversarial dataset against the baseline model BiDAF, it in general produces complex questions;
- It requires a complex linguistic understanding on the paragraph to produce a correct answer. This is backed by the statistic that an average of 2.18 spans are needed to answer the question with the average distance between them 26 words.
- It has a diverse pool of span questions and many counting and arithmetic questions, which encourages discrete reasoning.

These characteristics are also the reason why this task is important; a model that performs well on DROP would necessarily obtain a good semantic understanding as well as basic discrete reasoning.[1]

2 Methodology

In view of the difficulty of the dataset due to its various reasoning capability requirements, we decide to focus on solving only parts of the dataset that either have answer as set of spans within the context or as integer number from 0 to 9.

We mainly follow the architecture of QANet to construct our model, except that after obtaining the context-query attention, we use a feed-forward layer to predict the above mentioned two possible answer types. The span answer prediction still follows that of the QANet, i.e. using three encoders to predict the positions of the answer within the paragraph, while for the number prediction, we feed the attention output to a layer composed of a feed-forward network before outputting a single numeric answer. Our model is built with reference to the NAQAnet architecture (Dua et al., 2019) and has indeed smaller capacity than NAQAnet. Below we formulate our problem more precisely and introduce different componenets of our architecture, and the code and data are available on OneDrive.

[1] The dataset is found to contain little bias by a heuristic baseline, so it is basically unlikely to find loopholes to perform well.

Reasoning	Passage Highlights	Question	Answer	BiDAF
Subtraction (31.2%)	That year, his Untitled (1981), a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for $16.3 million, well above its $12 million high estimate.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	4300000	$16.3 million
Comparison (20.4%)	In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court . . . . In May 1518, Charles traveled to Barcelona in Aragon.	Where did Charles travel to first, Castile or Barcelona?	Castile	Aragon
Selection (18.4%)	In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack to tell the story of the events that led up to the battle.	Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?	Don Mueller	Baker
Addition (12%)	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near ˇ Sibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battlegroup to counterattack the next day.	What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured?	3 March 1992	2 March 1992
Count (16%) and Sort (8.8%)	Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal, yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal. . . . Carolina closed out the half with Kasay nailing a 44-yard field goal. . . . In the fourth quarter, Carolina sealed the win with Kasay's 42-yard field goal.	Which kicker kicked the most field goals?	John Kasay	Matt Prater
Coreference Resolution (4%)	James Douglas was the second son of Sir George Douglas of Pittendreich, and Elizabeth Douglas, daughter David Douglas of Pittendreich. Before 1543 he married Elizabeth, daughter of James Douglas, 3rd Earl of Morton. In 1553 James Douglas succeeded to the title and estates of his father-in-law.	How many years after he married Elizabeth did James Douglas succeed to the title and estates of his father-in-law?	10	1553
Other Arithmetic (2.8%)	Although the movement initially gathered some 60,000 adherents, the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75%.	How many adherents were left after the establishment of the Bulgarian Exarchate?	15000	60,000
Set of spans (2.4%)	According to some sources 363 civilians were killed in Kavadarci, 230 in Negotino and 40 in Vatasha.	What were the 3 villages that people were killed in?	Kavadarci, Negotino, Vatasha	Negotino and 40 in Vatasha
Other (6.4%)	This Annual Financial Report is our principal financial statement of accountability. The AFR gives a comprehensive view of the Department's financial activities ...	What does AFR stand for?	Annual Financial Report	one of the Big Four audit firms

Table 1: Representative examples from the DROP dataset illustrating different phenomena.

2.1 Problem Formulation

The input of the reading comprehension task consists of a context paragraph with n words $C = \{c_1, c_2, ..., c_n\}$ and a query sentence with m words $Q = \{q_1, q_2, ..., q_m\}$. Our goal is to output a span $S = \{c_i, c_{i+1}, ..., c_{i+j}\}$ from C or a number from $\{0, 1, ..., 9\}$ based on the predicted answer type (span or number).

2.2 Model Overview

Our model contains the original five components from the architecture of QANet (Yu et al., 2018): an embedding layer for the context and the query, an embedding encoder layer, a context-query attention layer, a model encoder layer and an output layer for the span answer prediction. Additionally, we include a linear layer for answer type prediction and another linear layer for count answer prediction, and the whole model is as shown in Figure 1.

Input Embedding Layer Each word of the context and the query are embedded on the word level and on the character level, which are subsequently transformed and concatenated to represent the word x. The word level embedding $x_w \in \mathbb{R}^{300}$ is fixed by the 300-dimensional pre-trained GloVe word vectors, and one thing to note for our model is that the <UNK> embedding is not set to be trainable,

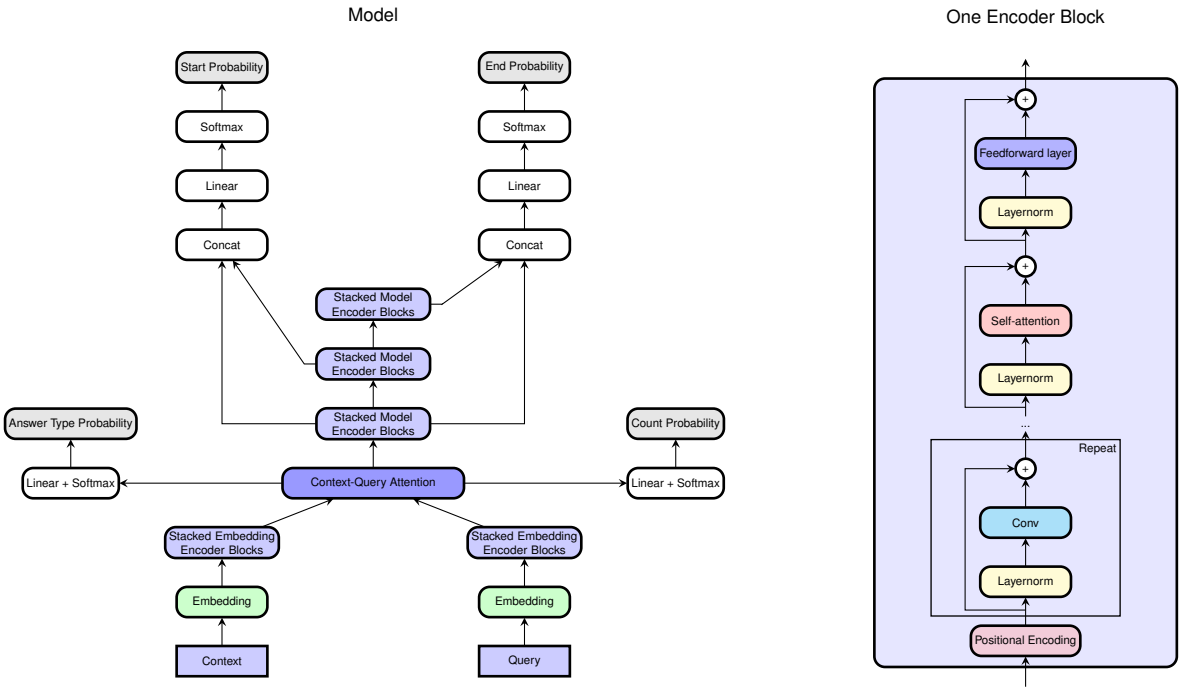

Figure 1: An overview of our model, built on the QANet model. The same encoder block (right) is used throughout the model. Weights are shared between the context and query embedding, and between the three output encoders (in practice, we just instantiate each class once and use the same instance at different places). For positional encoding, different positions receives different log time scales, and *sin* and *cos* functions are applied to obtain the encoding.

either. For character embedding (Kim et al., 2015), each character is embedded as a 200-dimensional trainable vector and each word is truncated or padded to 16 characters, which gives us a 200×16 matrix for each word. A 2-d convolutional network (input channel 200, output channel 128) is then used to transform this matrix to a 128×16 matrix, and the row maximum is taken to yield a 128-dimensional vector representation $x_c \in \mathbb{R}^{128}$ for the word. We then apply a 1-d convolutional network (input channel 300 + 128, output channel 128) to the concatenation of x_w and x_c to obtain a 128-dimensional representation $x \in \mathbb{R}^{d=128}$. On top of this representation, a two-layer highway network (1-d convolutional network, input channel 128, output channel 128, kernel size 1, stride 1 and no padding) (Srivastava et al., 2015; Seo et al., 2018) is applied to output the final 128-dimensional embedding vector $x \in \mathbb{R}^{d=128}$.

Embedding Encoder Layer The building block of embedding encoder layer consists of 4 1-d convolutional layers, a self-attention layer and a feed-forward layer, all of which interleaved with a layer-norm and placed inside a residual block, as shown in the right of Figure 1. For the convolutional layer, depthwise separable convolutions (input channel 128, output channel 128, kernel size 7) are used, which means that each channel is convolved with one filter to output a channel and no summation is done between different channels. For the self-attention layer, a multi-head attention mechanism defined in (Vaswani et al., 2017) with 8 heads is used. Note that the individual word of input and output of this layer are both 128-dimensional vectors. The number of encoder blocks for this layer is 1.

Context-Query Attention Layer The context-query attention layer takes in the embedded context $C \in \mathbb{R}^{n \times d}$ and query $Q \in \mathbb{R}^{m \times d}$, whose words are 128-dimensional vectors respectively. A similarity matrix $S \in \mathbb{R}^{n \times m}$ is then computed pairwise between C and Q with a trilinear function:

$$S(q, c) = W_0[q, c, q \odot c].$$

A softmax is then applied to each row of S to obtain a matrix $\bar{S}$, and the context-to-query attention is subsequently computed as $A = \bar{S} \cdot Q^T \in \mathbb{R}^{n \times d}$, which represents a query-aware context. Furthermore, a column softmax-normalized matrix $\bar{\bar{S}}$ is used to obtain the query-to-context attention $B = \bar{S} \cdot \bar{\bar{S}}^T \cdot C^T \in \mathbb{R}^{n \times d}$. The layer then returns $C, A, C \odot A, C \odot B$, which are all in $\mathbb{R}^{n \times d}$.

Model Encoder Layer The outputs from the context-query attention layer are first concatenated and resized from $4d$ to d before taken as input for the model encoder layer. The layer shares the same architecture as the embedding encoder layer, with the number of convolutional layers set to 2 and kernel size set to 1, and the total number of encoder blocks set to 7. The encoder layers are repeated 3 times to obtain representations M_1, M_2 and M_3 of the context-query input, and they all belong to $\mathbb{R}^{n \times d}$.

Answer-Type Output Layer This layer first uses some trainable weights to calculate vectors $h^C \in \mathbb{R}^d$ and $h^Q \in \mathbb{R}^d$ representing information found in the query-aware context $A \in \mathbb{R}^{n \times d}$ and the query $Q \in \mathbb{R}^{m \times d}$, respectively:

$$\alpha^C = \text{softmax}(W^C \cdot A),\ h^C = \alpha^C \cdot A,$$
$$\alpha^Q = \text{softmax}(W^Q \cdot Q),\ h^Q = \alpha^Q \cdot Q.$$

The two vectors are then concatenated and input into a feed-forward layer followed by a softmax to predict the answer type:

$$p^{\text{type}} = \text{softmax}(\text{FFN}([h^C, h^Q])).$$

Span-Answer Output Layer This layer predicts positions of the answer within the context for the span answer type. The probabilities of the starting position and ending position of the answer are calculated as:

$$p^1 = \text{softmax}(W_1[M_1; M_2]),$$
$$p^2 = \text{softmax}(W_2[M_1; M_3]).$$

Count-Answer Ouput Layer To perform counting on the context, only the representation vector h^C is used:

$$p^{\text{count}} = \text{softmax}(\text{FFN}(h^C)).$$

3 Experiments

In this section, we analyze some statistics of the dataset and introduce our experimental settings.

3.1 Dataset and Experimental Settings

Upon restricting our focus on the span and number answer type, the dataset question answer pairs shrink from 77k to 50k. The number of words of the contexts and the questions of the training set are plotted in Figure 2. We see that the length of the contexts is under 500 words, and under 30 for questions. We therefore set the context length limit to 400 tokens and question length limit to 50 tokens. This further reduces the question answer pairs to 46k, and among them 26,025 are of span answer type and 19,854 are of number answer type.

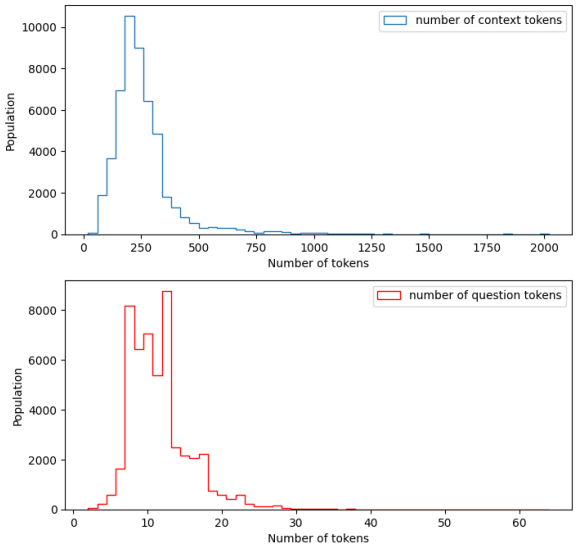

Figure 2: Number of tokens of the contexts and the questions in the training set

We use the cross entropy loss during training, which is composed of loss on answer type prediction, span answer prediction and number answer prediction. One important thing to note is that for the span answer prediction, the training loss is the sum of losses from all span answers in the set of spans, in line with the setting in Dua et al. (2019); Clark and Gardner (2018). This encourages the model to find spans from multiple positions within the context, even though the final answer just yields two positions.

We use Spacy tokenizer to tokenize the contexts and questions. On the hyper-parameters, we use L2 weight decay on all trainable parameters, with $\lambda = 3 \times 10^{-7}$. We also use dropout on the word, character embeddings and in between different layers, with dropout rate 0.1 for word embedding, 0.05 for character embedding and 0.1 between layers. The hidden size throughout the model is 128, and the batch size is 32. Adam optimizer with $\beta_1 = 0.8, \beta_2 = 0.999, \epsilon = 10^{-8}$ is used. The model is trained for 10 epochs.

For evaluation, the Exact Match (EM) and F1 score are used and the best score over all answers is recorded. Note that both EM and F1 would be recorded as zero if the predicted number answer is not equal to the ground truth.

3.2 Experimental Results

The exact match (EM) and the F1 score over the development dataset are 40.82 and 45.29, respectively. Upon examining the contribution from each answer type (span and count), however, we find significant disparity between the span and count answer types, with the EM and F1 score for the span answer type far higher than the number answer type. The results are summarized in Table 2.

Furthermore, upon close examination, we find that most count answers predicted are 2, and it prompts us to plot the number distributions in the training set and development set, which is shown in Figure 3. As can be seen, the model probably just predicts the most likely answer, 2, rather than performing discrete reasoning on the context and question to reach the answer.

Finally, we present some representative cases in Table 4, where we see that the model sometimes would attend to a wrong place in the context for extracting the span. In the fourth case, we would like to illustrate a caveat of the training loss we choose, which is that the model might correctly attend to different places of the context, but outputs a very long answer.

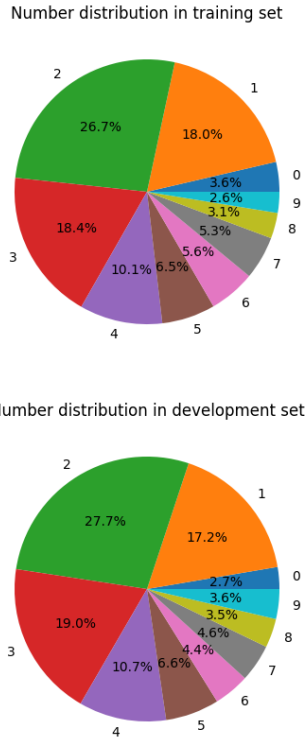

Figure 3: Number distributions in the training set and development set

Method	EM	F1
Heuristic Baselines		
Majority	0.07	1.44
Q-only	4.18	8.59
P-only	0.14	2.26
Semantic Parsing		
Syn Dep	9.48	11.33
OpenIE	9.26	10.33
SRL	10.87	13.35
SQuAD-style RC		
BiDAF	24.75	27.49
QANet	25.50	28.36
QANet+ELMo	27.08	29.67
BERT	29.45	32.70
Our Model		
P Span	47.29	54.37
Number	29.77	-
Complete Model	**40.82**	**45.29**
Human	92.38	95.98

Table 2: Performance of different models on the development set, in terms of Exact Match (EM) and token-level F1. Maximum score against the gold answers are chosen for both metrics. F1 score does not make sense for Count, and is not reported here.

3.3 Ablation Study

As an attempt to test the dependence of the model on different components or hyper-parameters, we have conducted ablation study on the answer type prediction and the character embedding, and the results are summarized in Table 3. We see that correctly predicting answer type gives an extra 1 point on the model performance, while surprisingly not training character embedding yields better results, which probably means that our model is overfitting on the training dataset.

4 Conclusion

In this project, we studied the DROP dataset, which is adversarial against the BiDAF model and requires different discrete reasonings to solve. We focus on the set of spans and number reasoning, extend the QANet architecture to predict answer types and make predictions on each answer type, and finally analyze experimental results and conduct some ablation studies. We find that our model performs reasonably in terms of set of spans prediction and the model does attend to different positions

Method	EM	F1
-train answer type prediction		
P Span	46.90	53.80
Number	29.56	-
Complete Model	40.49	44.85
-train character embedding		
P Span	48.11	55.14
Number	30.18	-
Complete Model	**41.49**	**45.92**
Our Model		
P Span	47.29	54.37
Number	29.77	-
Complete Model	40.82	45.29

Table 3: Ablation study results.

of the context as a result of the loss function setting, but for number prediction, the model likely picks up the statistics of the number distribution of the dataset instead of performing reasoning to reach the prediction. From the ablation study, we learn that predicting the answer type helps the model performance marginally and that our model likely overfits to the training set. DROP is indeed a challenging dataset that requires correctly predicting the discrete reasoning required and performs the reasoning on the context and question, and solving the dataset will help further develop the reasoning capability of natural language processing models.

Acknowledgements

Our model is built on the Pytorch QANet architecture implemented by Bang Liu, hengruo, andy840314 and hackiey.

References

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855, Melbourne, Australia. 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.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Bidirectional attention flow for machine comprehension.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks.

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.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Phenomenon	Passage Highlights	Question	Answer	Our model
Set of spans	... and wrote two treatises on book-keeping, The Maner and Fourme How to Kepe a Perfecte Reconyng (1553) and The Pathe Waye to Perfectnes (1569) ... His siblings included Anne (d. Jan 10, 1568/9), Isabel, Judith (d. Apr. 16, 1582) ...	In what year were Jame's Peele's treatises on book-keeping published?	1553, 1569	1568/9), Isabel, Judith (d. Apr. 16, 1582
Count + Sort	... the Lions flew south to play the Jacksonville Jaguars ... the Lions' Mikel Leshoure ran in three consecutive touchdowns, from 7, 1, and 8 yards out respectively to take a 21-point lead going into halftime ...	Which team scored more points in the first half?	Lions	Jaguars
Count	The price was high: the Russians had to pay 300,000 rubles and evacuate Kiev, Pereyaslav, Chernigov.	How many areas were the Russians forced to evacuate?	3	2
Set of spans	... followed by a 49-yard field goal by placekicker Matt Prater. The Rams responded, with quarterback Sam Bradford throwing a 36-yard touchdown pass to tight end Michael Hoomanawanui. In the second quarter, The Broncos took a 13-7 lead, with a 40-yard field goal by Prater.	What all field goals did Matt Prater make?	49-yard, 40-yard	49-yard field goal by placekicker Matt Prater. The Rams responded, with quarterback Sam Bradford throwing a 36-yard touchdown pass to tight end Michael Hoomanawanui. In the second quarter, The Broncos took a 13-7 lead, with a 40-yard

Table 4: Representative cases from our model on the development set.