Meta-Learning and Easy Data Augmentation for a Robust QA System

Zhiyin Lin,[1] Beining (Cathy) Zhou[1] Mentor: Allan Zhou

[1] Department of Computer Science, Stanford University

Stanford | ENGINEERING
Computer Science

Introduction

PROBLEM

Machine learning algorithms are mostly assessed on one fixed dataset split into train, validation and test sets. Yet, when applied to real world applications, data can often be distributed differently. In order to address that problem, we try to improve the robustness of our models to increase its generalizability.

Question Answering systems has shown great success in answering **in-domain** queries, but similar to many NLP problems, it suffers from poor generalizations in few-shot **out-of-distribution** contexts.

Meta learning is an approach that "learns to learn" from other ML models. It shows promise in effectively generalizing to out-of-domain data distributions. In short, it finds a sharp en-garde position for the model to attack a problem with few input data. Meta learning has been previously explored in few-shot NLP learning tasks. For instance, Dou. et. al has assessed the reptile model on the GLUE benchmark. However, meta learning in NLP has seen less success than in other fields. We explore the possibility of implementing meta learning to out of domain question answering.

In addition, data augmentation methods have shown impressive results in improving the robustness, particularly in computer vision. Thus, we explore various augmentation operations and investigate the best combination of operations and hyperparameters to boost performance.

DATASETS

Dataset	Question Source	Passage Source	Train	dev	Test
		in-domain datasets			
SQuAD [5]	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA [7]	Crowdsourced	News articles	50000	4,212	-
Natural Questions [6]	Search logs	Wikipedia	50000	12,836	-
		oo-domain datasets			
DuoRC [9]	Crowdsourced	Movie reviews	127	126	1248
RACE [10]	Teachers	Examinations	127	128	419
RelationExtraction [11]	Synthetic	Wikipedia	127	128	2693

Table 1. Statistics of 3 in-domain datasets and 3 out-of-domain datasets (borrowed from CS224N Project Instruction)

An Example Datapoint

{"**title**": "Leonardo Ghiraldini (born 26 December 1984 in Padu",
"**paragraphs**": [{"**context**": "leonardo ghiraldini born dec in padua is an italian rugby union player for leicester tigers in the aviva premiership ",
"**qas**": [{"**question**": "Which team does Ghiraldini play for?",
"**id**": "7bdeeb1e7cba4b64b0d981e603e17b50",
"**answers**": [{"**answer_start**": 90,
"**text**": "Leicester Tigers"}]}]}]}

BASELINES

DistilBERT: 66 million parameters, a smaller, distilled version the original BERT (~340 million parameters) + AdamW optimizer

- batch_size = 16
- learning_rate = 3e-5
- Train: epoch # = 3, eval per 2000 steps
- Finetune: epoch # = 5, eval per 10 steps

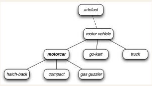
	Pretrain	Train	Train_Val	Finetune	Finetune_Val	Test
Baseline	DistilBERT	ID_train	ID_val	/	/	OOD_test
Baseline-Finetune	DistilBERT	ID_train	ID_val	OOD_train	OOD_val	OOD_test

Table 2. Experimental Pipeline used by Baselines

Methods and Experiments

EASY DATA AUGMENTATION

Data pre-processing
- All words lowercase
- Numbers exempted
- Punctuations removed



EDA Operations
- **Synonym Replacement**: from synonym dictionary NLTK WordNet
- **Random Insertion**: avoids stop words such as "she", "and", "at", etc.
- **Random Swap**
- **Random Deletion**

Example: Augmented Contexts

Operation	Context
Original	Ray Eberle died of a heart attack in Douglasville, Georgia on August 25, 1979, aged 60.
Synonym Replacement	beam eberle died of a heart flack in douglasville georgia on revered cured.
Random Insertion	ray along pass away eberle died of a heart hoosier state attack in douglasville georgia re on august aged.
Random Swap	august on a died of heart attack in douglasville georgia eberle ray aged .
Random Deletion	ray a heart attack douglasville august.

Table 3. Contexts generated by EDA operations. Example taken from Relation Extraction dataset. Each operation has α = 0.3

	Pretrain	Train	Train_Val	Finetune	Finetune_Val	Test
EDA	DistilBERT	ID_train	ID_val	OOD_train_eda	OOD_val	OOD_test

Table 4. Experimental Pipeline used by EDA Methods

EDA ABLATION STUDY

Hyperparameters:
- **α = {0.05, 0.1, 0.2, 0.3, 0.4}**
  - Percentage of words changed by each EDA method

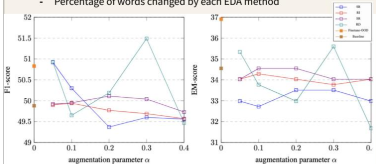


Figure 1. F1-score and EM-score of four EDA operations, Finetune-OOD, and Baseline. SR: synonym replacement. RI: random insertion. RS: random swap. RD: random deletion

- **N_aug = {1, 2, 4, 8, 16}**
  - Number of augmented contexts generated per original context

N_{aug}	1		2		4		8		16	
Operation	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
SR-alpha-0.05	50.92	32.98	50.32	***34.55***	**51.33**	34.03	49.98	30.63	50.44	31.94
RI-alpha-0.1	49.94	***34.29***	**50.76**	32.46	49.97	***34.29***	49.52	33.77	50.37	34.03
RS-alpha-0.2	50.11	34.55	**50.58**	***34.82***	49.58	33.25	49.95	34.29	50.37	33.25
RD-alpha-0.3	**51.49**	***35.60***	49.69	31.94	49.43	33.51	50.56	34.82	50.04	34.55

Table 5. F1-score and EM-score of N_aug = {1,2,4,8,16} for the four EDA operations, each with best performing augmentation parameter α. Optimal performance of each method is bolded for F1-score and italicized for EM score. The best performance across all experiments is marked in red.

META LEARNING SETUP

Meta Learning
- "Learn to learn"
- Effective for few-shot environments

Reptile
- Seeks a good initialization of a neural network so that the model could be easily fine-tuned on few-shot datasets

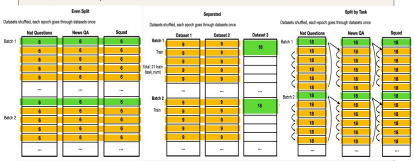
```
Algorithm 1 Reptile (serial version)
Initialize ϕ, the vector of initial parameters
for iteration = 1, 2, ... do
    Sample task τ, corresponding to loss L_τ on weight vectors ϕ̃
    Compute ϕ̃ = U_τ^k(ϕ), denoting k steps of SGD or Adam
    Update ϕ ← ϕ + ε(ϕ̃ − ϕ)
end for
```

META LEARNING EXPERIMENTS

Hyperparameters
- inner_lr: 0.1 (meta learning rate)
- n_inner_iter: 5 (number of inner loop iterations for gradient update)
- task_num: 21 or 3 (number of iterations)
- sample_size: 18 (number of data used in a task)

Data Split
- Meta learning usually takes in a wide range of different asks
- Each batch uses the same validation data

1. **Even Split**: each task contains both train and val samples evenly from 3 in-domain datasets
   The meta update step will see equal amount of data from each dataset, analogous to the OOD train set
2. **Separated**: train samples evenly from 2 in-domain datasets, val samples from the third in-domain dataset (Natural Questions)
   The update step will see data from two ID distributions. The third dataset is analogous to an OOD
3. **Split by Task**: each task only contains data from 1 dataset, task_num=3, and the training rotates among the datasets
   Each task is different because it covers a different dataset. The model should learn to adapt after several iterations to "learn to learn".



- **Inner Iteration Data**
1. Repeated: each task uses the same group of data
2. Different: each inner iteration uses a different data. We wanted each task to see more data (18 is largest for the machine). The model would see 5x more data and train 5x faster.

Results

Datasplit	Inner Iteration	F1 Score	F1 - Finetune	EM Score	EM - Finetune
Even Split	Repeated	**44.04**	**45.93**	**29.06**	30.89
	Different	38.79	43.53	21.73	29.06
Separation	Repeated	43.25	45.58	26.18	**31.68**
	Different	35.71	43.33	19.11	29.58
Split by Task	Repeated	16.71	35.8	7.07	23.56

Table 6. Results from meta learning models trained with Even Split, Separation, and Split by Task data splits.

Discussion

DISCUSSION

Data Augmentation
- **Alpha (α)**
  - Too much augmentation (α = 0.4) hurt performance – may have changed the meaning or sentence structure
  - Random Insertion + Random – flat curves
    - These methods do not omit or alter meaning, only add to it
  - Random deletion – rocky curve
    - Depend largely on which words are deleted by random chance
- **N_aug**
  - Most methods performed best at N_aug = 2 or 4
    - Larger N_aug may have caused too much conflicting confusion
  - Random deletion peaked at N_aug = 1
    - Deletion could significantly alter the meaning of a sentence
- **F1 Score**
  - Highest performance achieved by Random Deletion
    - F1 = 51.49 at α = 0.3
- **EM Score**
  - The trend of EM score matches that of F1, but consistent perform below finetune OOD
  - Obtaining exact match may be harder after perturbing original data

Meta Learning
- **Data Splits**
  - **Even split**
    - Best performance
    - When the tasks at hand are not diverse (three different datasets), it may be better to mix them to achieve a higher performance
  - **Separation**
    - Performance improved drastically during finetune
    - High EM scores
  - **Split by task**
    - Worse performance
    - Drastic improvement during finetune
- **Inner Iteration Data**
  - Repeated feeding is slow (takes roughly 10 hours)
  - Different feeding speeds up by 5 fold, but decreases performance significantly
    - Model depends on some degree of repetition
- **Overall**, model may have some success in finding good initialization, yet still could not beat baseline
  - Out-of-distribution may be too similar to in-distribution for meta learning to fully unleash its power in domain adaptation

FUTURE DIRECTIONS

Data Augmentation
- Regarding rocky behaviors observed in EDA, is the operations more effective on some words (parts of speech, position in sentence, etc) than other? For example, does deleting a verb or a proposition have the same effect?
- What would happen if we augment the questions, or even the answers?

Meta Learning
- For Data Split - Separated, compare performance when each of 3 ID dataset as the third dataset. Since each is of a different passage source, "Wikipedia" ones might be more effective for OOD (also has "Wikipedia")/
- With more diverse in-domain tasks, would meta learning yield better results?
- If given fewer out-of domain data to adapt to, or if given more shifted distributions, would meta learning perform better than the baseline?