Exploring Fluent Query Reformulations with Text-to-Text Transformers and Reinforcement Learning

Jerry Zikun Chen[1,2], Shi Yu[2], Haoran Wang[2]

[1]Department of Computer Science, University of Toronto
[2]Center for Analytics and Insights, The Vanguard Group

Contributions

Training text-to-text transformers (T5) with reinforcement learning (RL) to reformulate queries by adapting to F1 reward from downstream environment (question-answering, intent classification)

- Fine-tuned T5 is more sample efficient for RL than previous approach AQA, and generates more fluent reformulations.
- Use a separate T5 model to evaluate well-formedness of produced queries.
- Flexible framework adapted to other environments like intent classification when rewards can be engineered.

Methodology

Given a query q^i, a reformulation $r^i = \{r_1^i, \cdots, r_t^i\}$ is generated by Seq2Seq policy π_θ to compare with the target sequence $y^i = \{y_1^i, \cdots, y_t^i\}$.

Supervised Fine-tuning: Log-likelihood loss $-\sum_{i=1}^{N} \log \pi_\theta(\mathrm{r}^i|\mathrm{q}^i)$

$$= -\sum_{i=1}^{N}\sum_{t=1}^{T} \log p(r_t^i|r_1^i, \cdots, r_{t-1}^i, \mathrm{q}^i)$$

$$= -\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{V} y_{j,t}^i \log p(r_{j,t}^i|r_1^i, \cdots, r_{t-1}^i, \mathrm{q}^i) \text{ (i.e. cross-entropy loss)}$$

where V is vocabulary size and $y_{j,t}^i \in \{0, 1\}$ is the binary label at time t for token j and $p(r_{j,t}^i|\cdot)$ is the conditional probability of sampled token j at time t.

Reinforcement Learning:

$$\mathcal{J} = \sum_{i=1}^{N} \mathbb{E}_{r_t^i \sim \pi_\theta}\left(\sum_{t=1}^{T} R(r_1^i, \cdots, r_t^i)\right)$$

where R is the enviornment's reward function. $\nabla\mathcal{J}$ can be estimated by sampling reformulation r^i from π_θ (policy gradient):

$$\nabla\mathcal{J} \approx \sum_{i=1}^{b} \nabla_\theta \mathrm{r}^i \log \pi_\theta(\mathrm{r}^i|\mathrm{q}^i)(R(\mathrm{r}^i) - B^i)$$

which is the gradient of a weighted cross-entropy loss with sampled tokens as labels. B^i is the baseline for variance reduction. Additionally, scaled entropy $\lambda H(\pi_\theta) = \lambda \sum_t \sum_j p(r_{j,t}|r_{<t}, \mathrm{q}^i) \log p(r_{j,t}|r_{<t}, \mathrm{q}^i)$ is added to mitigate deterministic policy updates.

SearchQA Reward: QA dataset is SearchQA with context, query, answer triplets. The reward from a fixed BiDAF QA environment is given by the character-level F1 score

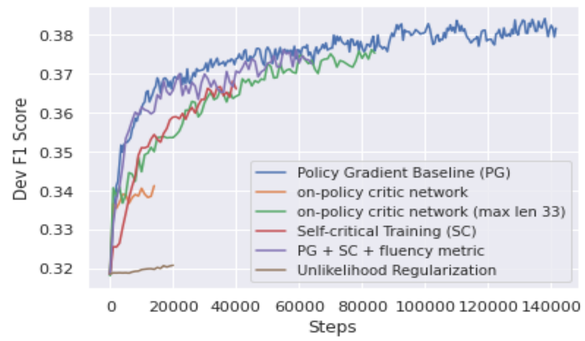
$$R_{F1} = 2\frac{p \cdot r}{p + r}$$

where p is precision and r is recall between true answer and the answer produced when reformulation is used. Validation (dev) set F1 score shows how well models can generalize on unseen set of data during RL.

AQA Framework

- Supervised pre-training: multilingual translation (UN Parallel Corpus) and paraphrasing (Paralex) (starting point for below RL methods)
- Baseline RL method: B^i is mean reward in the batch
- Value network: $B^i = f_c(\pi_\theta^E(\cdot|\mathrm{q}^i))$, f_c is a two-layer critic network on top of encoder π_E
- Self-critical training: $B^i = R(\mathrm{r}^i_{greedy})$, $\mathrm{r}^i_{greedy} \sim \pi_\theta(\cdot|\mathrm{q}^i)$
- Unlikelihood loss: $\mathcal{L}_{UL} = \sum_{t=1}^{T}[-\alpha \sum_{c \in \mathcal{C}^t} \log(1 - p(c|\mathrm{r}_{<t})], \mathcal{C}^t = \{r_1, \cdots, r_{t-1}\} \backslash \{r_t\}$
- Fluency metric: $R_f = \frac{1}{1+H(r)}, H(r) = -\frac{\sum_{t=1}^{T} \log p(r_t|\mathrm{r}_{<t}, \mathrm{q})}{T}$
- **Baseline method performs best for reward acquisition**

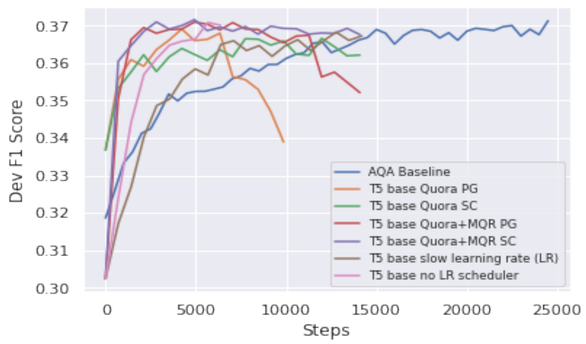
AQA SearchQA Dev Set F1 Reward Curve



T5 Framework

- Self-supervised pre-training: HuggingFace T5-base model pre-trained on C4 dataset
- Supervised fine-tuning: paraphrasing (Quora) and denoising (MQR) with prefix *"paraphrase: "* (starting point for below RL methods)
- Baseline RL method: B^i is mean reward in the batch
- Self-critical training: $B^i = R(\mathrm{r}^i_{greedy})$, $\mathrm{r}^i_{greedy} \sim \pi_\theta(\cdot|\mathrm{q}^i)$
- **RL with T5 is more sample efficient than AQA**

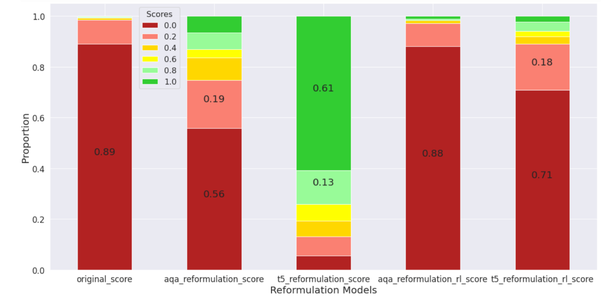
T5 SearchQA Dev Set F1 Reward Curve



Well-formedness

- Fine-tune a T5-base model on question well-formedness to quantitatively evaluate the fluency of produced reformulations (*"query wellformedness: "* prefix)
- **RL hurts query fluency for both T5 and AQA models, while T5 model after RL can retain more fluency compared to AQA**

Distribution of Predicted Well-formedness Scores on SearchQA Validation Set



Qualitative Examples

Original Text	T5 Reformulations (no RL)	T5 Reformulations (RL)	AQA Reformulations (no RL)	AQA Reformulations (RL)
Process for One time wire	What are the best methods for creating one time wire?	What is the process for a one time wire?	Process for one time wire?	What is one time wire process for one time wire name name one time...
this work object hasnt been touched in two weeks and is just sitting there	What can I do to solve this working object that hasn't been touched in two weeks and is sitting there?	What to do about a work object that hasn't been touched in two weeks and is just sitting there?	This work object has been touched in two weeks and just sitting?	What is this work object hasnt been touched in two weeks and just sitting have just name?
how to reinvest for an account?	How to fund reinvestments to an account?	How do I reinvest for an account?	How do you reinvest a account?	What is how to reinvest a account?

Comparisons on Internal Query Dataset between AQA and T5

Intent Classification

- Reformulator receives reward based on true/partial intent class matching in RL, and improves pre-trained BERT intent classification accuracy by 2% on internal dataset
- **Though supervised learning can improve fluency drastically, RL may be needed to correct the course for semantic drift to preserve intent**

Related Works

Buck et el. "Ask the Right Questions: Active Question Reformulation with Reinforcement Learning" (ICLR, 2017)
Raffel et el. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (JMLR, 2020)
Yu et el. "AVA: A Financial Service Chatbot based on Deep Bidirectional Transformers" (arXiv, 2020)