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, Toronto, Canada
[2]Center for Analytics and Insights, The Vanguard Group, Valley Forge, U.S.A.
jzchen@cs.toronto.edu, {shi_yu, haoran_wang}@vanguard.com

Abstract

Query reformulation aims to alter potentially noisy or ambiguous text sequences into coherent ones closer to natural language questions. In this process, it is also crucial to maintain and even enhance performance in a downstream environments like question answering when rephrased queries are given as input. We explore methods to generate these query reformulations by training reformulators using text-to-text transformers and apply policy-based reinforcement learning algorithms to further encourage reward learning. Query fluency is numerically evaluated by the same class of model fine-tuned on a human-evaluated well-formedness dataset. The reformulator leverages linguistic knowledge obtained from transfer learning and generates more well-formed reformulations than a translation-based model in qualitative and quantitative analysis. During reinforcement learning, it better retains fluency while optimizing the RL objective to acquire question answering rewards and can generalize to out-of-sample textual data in qualitative evaluations. Our RL framework is demonstrated to be flexible, allowing reward signals to be sourced from different downstream environments such as intent classification.

Introduction

Query reformulation and paraphrase generation techniques are employed for a variety of purposes in natural language processing (NLP), such as dialogue generation (Liu et al. 2018), machine translation (Madnani and Dorr 2010), and especially in question answering (QA) systems (Figueroa and Neumann 2013; Witteveen and Andrews 2019; Elgohary, Peskov, and Boyd-Graber 2019). Generating coherent and clean texts can reduce potential errors in downstream systems. In the cases when users are at the receiving end of NLP pipelines, it is essential to show them fluent and human-like languages before they lose faith and recede into requiring human agents for the sake of better understanding and communication. In search or question answering systems, query reformulation aims to paraphrase or restructure original question sequences, transforming them into ones that are more interpretable with natural well-formedness in both grammar and semantics. Typically, users may not have the patience to input an entirely grammatical or coherent question, which can cause issues for the downstream components to understand and give accurate predictions or answers. When human representatives are present, an originally noisy query or question can be reiterated and rephrased to double-check with users what they are asking for. This is a costly operation if every convoluted question needs to be restated. By having an NLP model to reformulate input queries, reformulations are fed back to users to confirm their original intentions in an automated way. As a result, unnecessary errors are eliminated and noises are prevented from propagating in an NLP pipeline, which can contain a series of models such as intent classification, information retrieval and question answering.

Traditionally, rule-based and statistical methods have been studied for paraphrase and reformulation generation (Meteer and Shaked 1988; McKeown 1979; Zhao et al. 2009). The advent of sequence-to-sequence learning (Seq2Seq) (Sutskever, Vinyals, and Le 2014) made it feasible to train deep neural networks as a new paradigm. We investigate how to paraphrase and denoise queries and generate well-formed reformulations using Seq2Seq learning models such as LSTMs (Hochreiter and Schmidhuber 1997) and transformers (Vaswani et al. 2017). Following the framework from AQA (Buck et al. 2018b), a Seq2Seq model is pre-trained on supervised tasks and further tuned using reinforcement learning (RL) on a machine comprehension QA dataset SearchQA (Dunn et al. 2017), learning from a pre-trained BiDAF (Seo et al. 2017) QA system that generates rewards. SearchQA is a suitable and challenging dataset as queries contain noisy phrases and the associated contexts are concatenated web text snippets from Google's search engine. Our goal is to obtain a model that can generate better-formed reformulations based on the original query sequences and achieve good QA performance with these reformulations. We use transfer learning (Ruder et al. 2019) from pre-trained transformers with text-to-text task formulations (Raffel et al. 2019). In our approach, pre-trained T5 models are first fine-tuned on paraphrase generation (Quora) and denoising (MQR) datasets to gain general paraphrasing

*Work done while interning at Vanguard
Association for the Advancement of Artificial Intelligence 2021 (www.aaai.org), Workshop on the 9th Dialog System Technology Challenge (DSTC-9)

capabilities. Then, reinforcement learning of downstream QA rewards is performed to further encouraged the model to produce task-specific reformulations. To our knowledge, this is a first attempt to fine-tune text-to-text transformers with RL, nudging the model to generate reward-acquiring query trajectories to get better answers. We show that fine-tuned text-to-text transformers are better starting points for RL as they are more sample efficient in achieving the same level of QA performance, acquiring rewards faster than the previous AQA approach that uses translation-based LSTMs. T5 models also generate reformulations with better readability and can generalize to out-of-sample data. We provide a new way to evaluate fluency on a sequence level using an trained metric on the well-formedness (QW) (Faruqui and Das 2018) dataset, which is based on real evaluations from humans, a more reliable source than widely-used algorithmic metrics based on overlapping n-grams.

Related Works

Our work is related to the task of paraphrasing. This is to restate a given sequence while preserving the same meaning. To use pre-trained language model's representation capabilities to generate paraphrases of sequences, Witteveen and Andrews (2019) focus on fine-tuning large language models with supervised datasets. USE (Cer et al. 2018), ROUGE-L (Lin 2004) and BLEU (Papineni et al. 2002) are measured to determine the best paraphrase. The models demonstrate the ability to generate paraphrases for out-of-sample sentences and paragraphs. Another common approach for paraphrasing is to leverage machine translation, Guo et al. (2019)'s work uses multilingual translation and pivoting for zero-shot paraphrases. Expensive human workers are employed in this process to evaluate fluency. Roy and Grangier (2019) uses a VQ-VAE (Oord, Vinyals, and Kavukcuoglu 2017) to compare a monolingual paraphrasing method with unsupervised and supervised translation-based approaches. Other generative approaches have also been explored for paraphrasing (Yang et al. 2019a; Gupta et al. 2018). Metrics based on n-gram overlaps are used in most of these works, even though it has been shown that BLEU or ROUGE do not agree well with human judgements (Callison-Burch, Osborne, and Koehn 2006; Stiennon et al. 2020). These N-gram based metrics require reference gold sentences, which are not available in the paraphrasing or reformulation task since there are many ways to reformulate the same query. Our evaluation of the reformulation qualities do not rely on gold references or any related algorithmic metrics. Fluency scores are generated by a T5-base model fine-tuned on the QW dataset containing human judgements.

Our reinforcement learning framework is closely related to Buck et al. (2018b)'s AQA approach by leveraging policy-based RL methods to generate question reformulations. These reformulations are treated as inputs to a BiDAF (Seo et al. 2017) question answering system that generates token-level F1 rewards. However, their GNMT (Wu et al. 2016) reformulation model depends on complex pre-training on multilingual translations and paraphrasing procedures that are not reproducible from its open-sourced project. Our approach leverages recent advances in Seq2Seq learning and transfer learning, enabling direct fine-tuning of a pre-trained text-to-text transformer with paraphrasing and denoising tasks. This gives flexibility in what starting point we can use before we enter the RL stage. The general linguistic and reformulation knowledge encoded in the model helps retain sequence-level fluency and well-formedness before and after RL training.

We leverage Text-to-text Transfer Transformers (T5) (Raffel et al. 2019) as the foundation of our reformulation and well-formedness models. As a systematic ablation study on transfer learning in NLP, this work compares and contrasts different transfer learning schemes extensively. The resultant best-performing model is T5, which has an encoder-decoder architecture unlike single-stacked BERT (Devlin et al. 2019) and its descendants (Liu et al. 2019; Yang et al. 2019b; Lan et al. 2020). The largest T5 model can achieve state-of-the-art results on many NLP benchmarks including SuperGLUE (Wang et al. 2019), where it reaches near human-level performance. T5 formulates any text-based NLP task in an unified text-to-text format, a natural fit for generative tasks like query reformulation. Fine-tuning task descriptions can be directly specified as a prefix to the input. This provides flexibility of fine-tuning on different tasks without having to change the training pipeline. We leverage the general linguistic knowledge of the English language implicitly encoded in the parameters of the transformer-based T5 model pre-trained on unlabeled text. We also train T5 further with policy-based RL after supervised fine-tuning. To our knowledge, using RL to tune a T5 model has not been attempted to our knowledge. (Lin et al. 2020) showed T5's superior performance on reformulating questions within a conversational history. Their model uses both the question and the context as input, whereas in our approach, only the original noisy query are used as input to the reward-generating black-box QA model at the RL stage. Identifying well-formed questions (Faruqui and Das 2018) by training binary classification models have been studied using BERT(Chhina 2020) and transfer learning with pre-trained models (Syed et al. 2019). Instead, we investigate more fine-grained 6-way classification using a fined-tuned T5 well-formedness model, leveraged as a proxy for evaluating sequence-level fluency of reformulators.

There has been a body of work in other domains to adopt RL for structured sequential prediction tasks. Keneshloo et al. (2020) survey various reinforcement learning techniques for training sequence-to-sequence models. In the context of abstractive summarization Keneshloo, Ramakrishnan, and Reddy (2019), they choose to use self-critical training (Rane, Sargar, and Shaikh 2018), which we leverage as an alternative method for RL. Stiennon et al. (2020) geneerate summaries with proximal policy optimization (Schulman et al. 2017) as the reinforcement learning algorithm with an KL-regularized reward signal. Angermüller et al. (2020) studies biological sequence generation with RL and points out the difficulty of value-based methods like DQN (Mnih et al. 2013) in the setting where rewards are episodic and delayed at the end of sequence generation, which is a similar setup to our task.

Methodology

Datasets

SearchQA (Dunn et al. 2017) is comprised of 140,000 question-answer pairs from the Jeopardy! archive. Contexts are generated by concatenating text snippets when questions are fed to Google's search engine. The queries are not in the form of proper questions. They appear convoluted and ambiguous, e.g. *"1909 Nobel prize winner failed entrance exams univ bologna , italy"*. The text snippets are noisy in a similar way. The task is to extract answers within the context given a question. Paralex (Fader, Zettlemoyer, and Etzioni 2013) is used by Buck et al. (2018b) to pre-train the reformulator. It contains question paraphrase clusters. 25,100 unique queries filtered from Paralex becomes the question well-formedness (QW) dataset (Faruqui and Das 2018), which we use for fine-tuning the well-formedness model. Every query in QW is examined by 5 human workers, judging whether the given query is ill-formed (a score of 0) or well-formed (a score of 1). Average scores among these 5 workers are reported as the well-formedness ratings. To fine-tune T5 models to gain paraphrasing and denoising abilities, Quora and MQR datasets are used respectively. The Quora dataset contains similar yet differently expressed question pairs from the Quora online Q&A forum. The multi-domain question rewriting (MQR) (Chu et al. 2020) dataset consists of ill-formed and well-formed question pairs, for example, *"Spaghetti carbonara, mixing"* is paired with *"How to mix a spaghetti carbonara?"*. Lastly, we leverage an internal dataset of 300k query logs produced by human agents in the financial QA setting for out-of-sample experiments.

Setup

Supervised Fine-tuning Given a query sequence $\mathbf{q} = \{q_1, q_2, \cdots, q_k\}$ with length k in a dataset of size N, the reformulator model produces a sequence of word distributions. A reformulation $\mathbf{r} = \{r_1, \cdots, r_T\}$ is produced by greedily sampling from these distributions at each time step. These words are matched with the target sequence $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$. We assume here that the target sequence has the same length as the reformulation. Practically, both reformulation and target sequence are padded to a default max length of 50, any word distributions produced after the end special token are disregarded in the loss. $\mathbf{q}$ and $\mathbf{y}$ consist of word pieces from a pre-defined vocabulary with size V produced by sentence-piece (Kudo and Richardson 2018). We use the default vocabulary of size 16,000 for AQA and 32,168 for T5. For the i'th data point, the conditional sequence probability of the reformulation $\mathbf{r}^i$ is given by:

$$\pi_\theta(\mathbf{r}^i|\mathbf{q}^i) = \prod_{t=1}^{T} p(r_t^i|r_1^i, \cdots, r_{t-1}^i, q_1^i, \cdots, q_k^i)$$

Among all N sequences, the i'th target sequence $\mathbf{y}^i$ is defined by one-hot correct labels at each time step. Therefore, the cross entropy loss between the model predictions $\mathbf{r}^i$ and target sequence is:

$$\begin{aligned}\mathcal{L}_{CE} &= -\sum_{i=1}^{N} \mathbf{y}^i \log \pi_\theta(\mathbf{r}^i|\mathbf{q}^i) \\ &= -\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{V} y_{j,t}^i \log p(r_{j,t}^i|r_1^i, \cdots, r_{t-1}^i, \mathbf{q}^i)\end{aligned}$$

where $y_{j,t}^i \in \{0, 1\}$ is the binary label at time step t for token j, and $p(r_{j,t}^i|r_1^i, \cdots, r_{t-1}^i, \mathbf{q}^i)$ is the conditional probability of token j appearing at time step t given by the reformulator model π_θ.

Note that minimizing cross entropy loss $\mathcal{L}_{CE}$ is equivalent to minimizing the negative log-likelihood $-\sum_{i=1}^{N} \log \pi_\theta(\mathbf{r}^i|\mathbf{q}^i)$

$$\begin{aligned} &= -\sum_{i=1}^{N}\sum_{t=1}^{T} \log p(r_t^i|r_1^i, \cdots, r_{t-1}^i, \mathbf{q}^i) \\ &= -\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{V} y_{j,t}^i \log p(r_{j,t}^i|r_1^i, \cdots, r_{t-1}^i, \mathbf{q}^i)\end{aligned}$$

Reinforcement Learning In the RL stage, we optimize for the expected long term rewards:

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

where R is the black-box reward function that generates rewards between 0 and 1 only at the end of generation $t = T$. In the question answering setting, our reward is the token-level F1 score between the true answer a and the generated answer a' when the reformulation is used as the input to a pre-trained BiDAF QA system. Precision p is the percentage of tokens in a' that are also in a. Recall r is the percentage of tokens in a that are also in a'. The F1 score is given by $R_{F1} = 2(p \cdot r)/(p + r)$. The main quantitative metric of interest is the F1 score on the dev set as this shows how well models can generalize on unseen set of data during RL. We do not check SearchQA test set scores because the pre-trained BiDAF QA environment provided by AQA is unable to find test set documents according to their IDs and produce corresponding scored answers. Therefore, the dev set is our sole evaluation set. Later, we check reformulation qualities of out-of-distribution internal dataset for further comparisons in qualities.

For a batch of size b, the gradient of $\mathcal{J}$ can be estimated by REINFORCE (Williams and Peng 1991), which is the policy gradient (Sutton et al. 1999) of the MLE objective weighted by the episodic reward produced by a sampled reformulation trajectory $\mathbf{r}^i = \{r_1^i, \cdots, r_T^i\} \sim \pi_\theta$ given an original query $\mathbf{q}^i$:

$$\begin{aligned}\nabla\mathcal{J} &\approx \sum_{i=1}^{b} \nabla_\theta \log p(r_1^i, \cdots, r_T^i|\mathbf{q}^i)(R(\mathbf{r}^i) - B^i) \\ &= \sum_{i=1}^{b} \nabla_\theta \mathbf{r}^i \log \pi_\theta(\mathbf{r}^i|\mathbf{q}^i)(R(\mathbf{r}^i) - B^i)\end{aligned}$$

This means that we can maximize the above weighted log likelihood function (i.e. minimizing weighted cross-entropy loss) as a surrogate to compute and estimate the policy gradient in each batch. The target sequences in this surrogate cross entropy loss are the greedily sampled reformulation trajectories from the policy for estimating the expectation. A trajectory that obtains a higher reward will produce a higher gradient signal to encourage generation of words close to this sampled reformulation at each time step. Similar to AQA (Buck et al. 2018b), for variance reduction, the mean reward of the minibatch is used as the baseline B in the gradient and the loss. In addition, a scaled entropy regularizer $\lambda H(\pi_\theta) = \lambda \sum_t \sum_j p(r_{j,t}|r_{<t}, \mathbf{q}^i) \log p(r_{j,t}|r_{<t}, \mathbf{q}^i)$ is added to the objective $\mathcal{J}$ to mitigate deterministic policy updates. The modifications in the next section are based on this objective function, we refer to it as the policy gradient (PG) baseline.

AQA Framework

We first follow and reproduce the AQA reformulation model. This LSTM reformulation model architecture is based on GNMT (Wu et al. 2016), pre-trained on multilingual translation and the Paralex dataset for general paraphrasing capabilities, as described in (Buck et al. 2018b). We directly use the pre-trained saved checkpoint as the starting model for the RL stage as pre-training procedures are unavailable. Note that we do not use the CNN selector from AQA as our focus is to produce a single reformulator that generates more fluent sequences while maintaining a decent QA performance. The same BiDAF model pre-trained on SearchQA is used as the black-box QA environment. The reformulator policy turns the input query into an reformulation, which is then passed to the QA environment obtaining a token-level F1 score. As mentioned, the baseline in this approach is the mean reward of the batch. A RL-trained reformulator with the same architecture provided by (Wu et al. 2016) is also investigated in terms of generation quality compared with our trained baseline PG model.

Reinforcement Learning The advantage actor-critic approach is a common modification for policy gradient methods where $A^i = R(r^i) - B^i$ is the advantage estimate for the i'th sample. We use a simple two-layer neural net f_c as an on-policy critic network to predict a value estimate for each batch. The output embedding from the GNMT encoder π_θ^E is used as the input for this critic network. The value becomes the baseline $B^i = f_c(\pi_\theta^E(\cdot|\mathbf{q}^i))$ and it is an estimation of the expected reward for the batch, given the current model's encoding of the input.

We test another common approach to tune Seq2Seq models, the self-critical training (Rane, Sargar, and Shaikh 2018). First proposed for the image captioning task, this method uses the reward generated by the greedy output $\mathbf{r}^i_{greedy} \sim \pi_\theta(\cdot|\mathbf{q}^i)$ as the baseline $B^i = R(\mathbf{r}^i_{greedy})$ in the policy gradient formulation. This is to encourage the model to outperform the greedy decoding strategy through trial and error.

Beside varying the RL algorithms, we implement methods that explicitly encourage fluency. Recent work (Welleck et al. 2020) show progress on remedying the problem where common tokens are over-generated and undesired repetitions cause issues in fluency. Unlikelihood training fits our task as reformulations produced by Buck et al. (2018b) often contain repetition of words and phrases. Unlikelihood training is proposed as an extra term to regularize the loss function and explicitly suppress the likelihood of negative candidate tokens $\mathcal{C}^t = \{r_1, \cdots, r_{t-1}\} \backslash \{r_t\}$ in a reformulation sequence $\mathbf{r} = \{r_1, \cdots, r_T\}$. In this method, the following loss is weighted by the advantage estimate with the mean reward as the baseline:

$$\mathcal{L}_{UL} = \sum_{t=1}^{T} [-\alpha \sum_{c \in \mathcal{C}^t} \log(1 - p(c|\mathbf{r}_{<t}) + \mathcal{L}_{MLE}]$$

$$\mathcal{L}_{MLE} = -\log p(r_t|r_{<t}, \mathbf{q})$$

Another addition is the fluency metric from Ge, Wei, and Zhou (2018), which is proposed for error correcting sequence generation and inference:

$$R_f = \frac{1}{1 + H(r)}, H(r) = -\frac{\sum_{t=1}^{T} \log p(r_i|\mathbf{r}_{<t}, \mathbf{q})}{T}$$

This metric ranges between 0 and 1 and incorporates the probabilities produced by the model. We use it as an extra reward signal on top of the F1 reward $R(\mathbf{r})$ from the QA environment.

T5 Framework

We first build the above mentioned modifications in AQA's framework and notice some difficulty with its implementations which leverages frameworks of GNMT (Wu et al. 2016) and BiDAF (Seo et al. 2017). Before the RL stage, the LSTM-based reformulator model is pre-trained on multilingual translation and paraphrasing, which are complex tasks that require specialized pipelines and datasets. Neither of these procedures are open-sourced for us to retrain from scratch with a different scheme. Thus, under the AQA framework, all methods start from a checkpoint already trained on multilingual translation and paraphrasing. Furthermore, the AQA reformulator has limited linguistic knowledge given that it is tuned for only two specific tasks before RL without any self-supervised pre-training. With the recent advances of transfer learning in NLP, we choose to leverage general language representation power encoded in a pre-trained transformer-based model. Due to these limitations of the AQA framework, we draw inspiration from Goutham (2020) and leverage the T5 model to replace the LSTM-based reformulator and use more flexible frameworks of Hugging Face (Wolf et al. 2019) and PyTorch Lightning (Falcon 2019). This gives us more flexibility to fine-tune on supervised tasks and create our own starting points for RL. T5 has a similar encoder-decoder structure as the original transformer (Vaswani et al. 2017), which is suitable for Seq2Seq tasks. The general and flexible setup formulates any language tasks into the text-to-text format, where a prefix description of the task is attached to each input, instructing the model to perform the task through text without having to vary much of the training pipeline.

Supervised Paraphrasing and Denoising Pre-training of sequence-to-sequence models before RL is a common practice (Buck et al. 2018b; Jaques et al. 2017). In a related work, Choshen et al. (2020) study RL for neural machine translation and show that having a pre-trained model close to producing good translation is crucial to obtain performance before RL is adopted. We follow these findings and train a new RL starting point to replace the machine-translation based model from AQA. This involves transfer learning of paraphrasing and denoising tasks with a T5-base model pre-trained on C4 corpus. This model has roughly 220 million parameters, similar to the capacity of the 210-million-parameter GNMT model. We first fine-tune the T5-base model on duplicated questions from the Quora dataset to achieve general paraphrasing ability. Then the MQR dataset is used to further fine-tune the model. The MQR work (Chu et al. 2020) suggests that Quora and MQR create a good combination for improving query qualities. This fits our purpose of transforming noisy queries into more fluent reformulations. For both Quora and MQR dataset, we fine-tune using the prefix *"paraphrase: "* for the input sequence and add "$<\backslash s>$" special token as a suffix to both the source and target sequences. We lightly train on the Quora and MQR datasets for two epochs respectively.

Reinforcement Learning Using the above pre-trained model, we adopt RL approaches similar to those in the AQA framework mentioned before. In particular, since the policy gradient (PG) baseline (with mean reward advantage and entropy regularization) shows the best QA performance in our results, we focus on this approach for T5 in RL training. Self-critical (SC) training, being another common approach with decent performance, is also used as an alternative method. These implementations are achieved by modifying the T5 module in Hugging Face (Wolf et al. 2019). Each module represents its own RL algorithm and can be swapped with flexibility. Since we want our T5 model to continue the reformulation task from the previous stage during RL, we add the same prefix and suffix to each input sequence as those in the paraphrase fine-tuning procedure mentioned above.

Query Well-formedness

For a non-algorithmic numeric proxy for sequence fluency unrelated, we use the QW dataset to fine-tune a separate T5-base model to create an automatic metric. Again, we cast regression as a text-to-text task by producing a text string of the average rating score and compare it against the true label, e.g. a score of 3.0 becomes *"3.0"*. The prefix for this task is *"query wellformedness: "*. Since the scores are averaged among 5 human workers, there are 6 possibilities from 0.0 to 1.0. Therefore, this can be seen as a 6-way supervised classification task in the text-to-text framework as mentioned in (Raffel et al. 2019), similar to the way another regression task STS-B is formulated. This model judges how well-formed or fluent a given query is based on human evaluations. The well-formedness T5 model is fine-tuned for 50 epochs on QW, when validation set accuracy no longer improves.

Intent Classification

To demonstrate the flexibility of our framework, we experiment with a different downstream black-box system, namely a pre-trained BERT-based intent classification (IC) model (Yu, Chen, and Zaidi 2020). We first use the supervised fine-tuned T5 model with no RL to reformulate every query naively in our internal intent classification dataset and train a new IC model with reformulations. We compare its performance to the original pre-trained model. Then, RL is applied to further adapt the reformulator to the IC system in a similar fashion as the QA system to reformulate queries to optimize for classification accuracy. Rewards are engineered in this way: when the predicted intent class exactly matches the true class, the reformulator receives a reward of one. Otherwise, if a match between the parent labels of the predicted class and true class occurs, the reformulator gets a partial reward of 0.5. If neither of these is matched, the IC system gives 0 reward. This is to show if interactive RL reformulations can be useful on an IC model that was pre-trained on the original noisy text, similar to the setup of QA.

Other Details

For all training variants, we use an Amazon EC2 g4dn.8xlarge instance with 32 CPU cores and a single NVIDIA T4 GPU. In all of the above mentioned RL stages, the pre-trained BiDAF QA environment runs on CPU cores and the reformulation model is trained on the GPU. We stop RL training when the reward curve stops improving significantly on the validation set.

Experimental Results

Reinforcement Learning

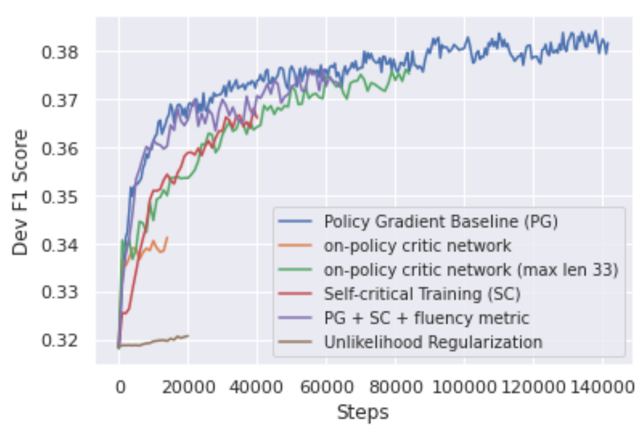

Figure 1: SearchQA Dev Set F1 Reward Curves with AQA

AQA Framework Figure 1 plots the mean dev (validation) set F1 scores on SearchQA during the RL training of the reformulator. The baseline method is the original AQA approach and the rest are variations that we mentioned in the previous section. The baseline AQA PG method takes 9 days to train on our single-GPU machine. Most of the variations are able to adapt to the RL objective and learn from token-level F1 scores through trial and error except for the unlikelihood objective. The unlikelihood method did not integrate with the model as well as we expected to reduce the number of repetitions and gain rewards. The addition of a critic network only surpasses the baseline initially and becomes flattened quickly. When we reduce the max length of the reformulation output produced by the translation model, the method with a critic network improves its performance. However, forcing the model to produce shorter reformulations is not ideal, the model must learn when to stop by itself. We notice for most of these methods including the best-performing PG baseline, long sequences tend to be produced. The lengths are often close to the default length restriction of 50 after RL. Furthermore, we observe that none of the changes we implemented for the RL algorithm can outperform the original baseline in learning the reward under the AQA framework. The closest reward curve to the baseline method is having a mixed loss function that combines the policy gradient and the self-critical objectives, with the addition of the fluency metric as an extra reward. In the following section, we experiment with the T5 modeling framework that can both retain fluency and optimize QA performance with RL. Among the above methods, SC training performs relatively well under the AQA framework and achieved the closest performance to the baseline when combined with the policy gradient objective, we adopt it as an alternative to the baseline PG method in the T5 framework.

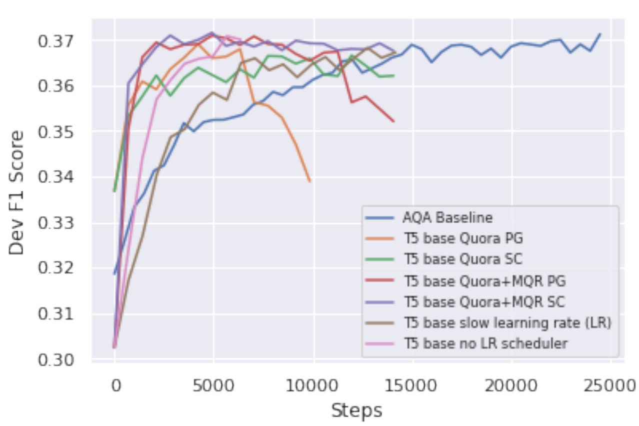

Figure 2: SearchQA Dev Set F1 Reward Curves with T5

T5 Framework In Figure 2, when T5 models are tuned with RL, we observe better sample efficiency with faster reward acquisitions. The starting checkpoint's reward on the validation set is lower than the AQA baseline model, but the reward grows quickly after a few epochs. The reward curves for policy gradient (PG) approaches eventually start to drop, and the self-critical (SC) approaches turn flat. In addition, when the gradient is tracked, its norm becomes large. We think this is due to RL algorithms taking steps in bad directions and not able to recover. Note that we are looking at the reward curves on the validation set rather than the training set to evaluate generalization. The reward curves on the training set do not drop, which means that overfitting can occur in the RL stage for T5 models. When the reward drops on the dev set, we observe the model produces more repetitions similar to AQA, losing query fluency. Changing the learning rate scheduler, entropy hyperparameters, or learning rates do not resolve this issue. Therefore, we pick the best-performing PG model with dev set F1 score around 0.37 as our model for qualitative analysis in the Appendix. Although the T5 model cannot train for longer without sacrificing rewards, it learns almost 3 times quicker than the GNMT model and reaches a comparable QA performance with only 0.01 difference in F1 as the AQA reformulator eventually reaches 0.38 dev F1 score after 9-days of training.

Score Category	Count	Accuracy	Average Absolute Difference
0.0	5360	0.720	0.099
0.2	4094	0.297	0.221
0.4	3018	0.173	0.294
0.6	2892	0.150	0.028
0.8	3751	0.157	0.018
1.0	5985	0.951	0.015

Table 1: Accuracies for Well-formedness Model

Well-formedness

We use QW dataset to fine-tune a T5-base model for quantitatively measuring fluency of queries. After fine-tuning, the well-formedness model achieves 42.32% for 6-way classification on the test set, which related works do not report. The well-formedness study (Faruqui and Das 2018) focus on binary accuracy using 0.8 as a threshold to determine whether a question is well-formed. Using the same threshold to group the multi-classed predictions, the binary classification accuracy is 79.56%, which is better than 70.7% reported by the best model in the original paper, higher than another transfer learning approach with 75.05% (Syed et al. 2019), close to the accuracy of 81.6% from a BERT-based model (Chhina 2020).

Accuracy for individual classes are presented in Table 1, we can see that scores 0.0 and 1.0 can be predicted with accuracy, but not for mid-range scores. However, the absolute differences are computed between each predicted score and gold score and then averaged across each class to get the last column, where we see that values are less than 0.3 for the mid-range classes. This means the model puts some questions into a neighbouring category so we believe this discrepancy is acceptable. Overall, we think this model is a decent human proxy to judge the coherence and fluency of reformulations even though there is room for improvement to reach the human upper-bound of 88.4% binary accuracy on the QW dataset reported by Faruqui and Das (2018).

With this well-formedness model to produce fluency scores, in Table 2 and Figure 3, we observe that when the T5 model (fine-tuned on Quora and MQR with no RL) is used to reformulate all queries in the dataset, the distribu-

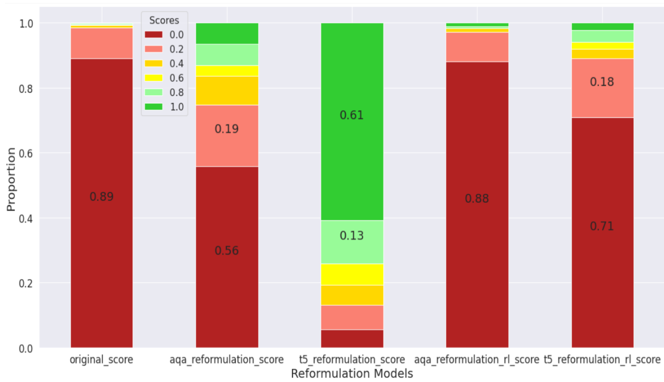

Figure 3: Distribution of Predicted Well-formedness Scores on SearchQA Validation Set

Reformulation Models	**Mean QW Scores of Reformulations on Search QA Validation Set**
No Reformulations	0.0275
T5 before RL	0.7933
T5 after RL	0.1128
AQA before RL	0.2106
AQA after RL	0.0381

Table 2: Accuracies for Well-formedness Model

tion of the well-formedness scores improve significantly. We then take models trained with PG as the RL algorithm to make fair comparisons on the fluency metric. We observe that RL hurts query fluency for both T5 and AQA models, the T5 model after RL can retain more fluency compared to its AQA counterpart. In addition, it is worth noting that the average QW score of 300k raw queries from our internal log dataset is 0.5015, compared to 0.0275 of SearchQA dev set, suggesting SearchQA is a challenging set in terms of fluency, at least more ill-formed than real-world questions in the financial Q&A setting.

Qualitative Comparisons

SearchQA Qualitatively, when given an original query, the reformulation qualities can vary for different models. For each approach, we generate reformulations with the best performing model on the validation set and decode greedily. Note that in Appendix Table 3, all models get the correct answer with an F1 score of 1. We see that the first model without RL does not understand what the question is asking for. Model 2 is the out-of-the-box RL-trained checkpoint downloaded from AQA. Model 3 is the PG baseline trained by us. Model 4 is the SC method. These three models are relatively more fluent, and model 4 does more exploration in picking the words but the reformulation gets less coherent. Model 5 mixes the PG and SC objective, which does relatively well on reward acquisition as shown in Figure 1, but it does not help the quality of this particular query even though the fluency metric is used as an extra reward. When T5 is used to reformulate the original noisy query in models 6 and 7 (before and after the PG method in RL), we see improvements in coherence. However, before the RL stage, it does not know what the purpose of this query is so we see it reformulates the query into a general "how" question based on its general knowledge from pre-training on a large corpus. After RL, it asks a "what" question with minimal expansion of the original sequence since it has been trained on reward signals indicating that these queries are normally asking for a certain kind of entity, in this case, the Nobel prize winner.

With Query 2, again, all models get the answer exactly with max F1 scores. Notice that this dataset consists of mostly ungrammatical and very noisy queries, similar to their concatenated snippets retrieved from the Google search engine. This is hard for our model to reformulate without having access to any context, even for us humans. Model 3 with PG baseline is the more fluent variant with less repetition than the downloaded model 2 even though the two approaches are the same. T5 models are constrained by the word *"funnier"* with this particular query, and the reformulation asks a yes-or-no question. However, the black-box BiDAF QA environment is still able to give the correct answers from these reformulations. Therefore, we note that as long as certain keywords are present, this QA system will give a good answer. This is also mentioned in the language analysis in (Buck et al. 2018a). However, in some cases, we note that the environment can never output the correct answer no matter how we try to reformulate queries as humans. For instance, as another example from SearchQA, a sequence of word salad: *"blue river runs black forest black sea"* is hard to be reformulated without any context. This can be stemmed from the idiosyncrasy of the particular QA environment, which is trained on the noisy original dataset. Although it may not be entirely reliable, we use this pretrained QA model for better comparing with the previous work.

Internal Dataset We sample from an internal textual dataset of client queries recorded by call representatives and evaluate AQA models and T5 models generalization on this out-of-domain dataset before and after RL training on SearchQA. In Table 4, for each input query, we generate multiple reformulations with beam search and picked ones we think have the best qualities. In the first three cases when original queries are fluent, T5 reformulators paraphrase the original query or make minimal changes. When the original queries are more ambiguous (in bold), we get reformulations that transforms and expands the noisy text into proper questions that are aligned with the original intention semantically.

In Table 5, we bring back AQA models for qualitative comparison. Compared to T5 models, it is clear that out-of-sample reformulations generated by AQA's GNMTs are more rigid and prone to repetitions and errors before and after RL on SearchQA, which is also corroborated in Table 3. This is addressed by Buck et al. (2018b) and Buck et al. (2018a). The reformulated language is regarded as an instance of machine-to-machine translation where repetitions are acceptable in the principles of information retrieval. However, based on qualitative analysis, we believe that AQA models that learn to communicate between machines are not adequate enough for production in the case when reformulations are required to be fluent on a level to be shown, examined and double-checked by human users in a QA setting.

Intent Classification

We find that training a new intent classification model with naive reformulations slightly hurts the performance in accuracy and F1 score. After RL, the pre-trained BERT IC accuracy can be improved by 2%. We think that even though supervised fine-tuning of reformulators before RL can improve the fluency of queries by a large margin as shown in Figure 3, the reformulations can drift away from the original intended purpose due to the discrepancy between the fine-tuning data and out-of-sample data, which in this case is our IC dataset. This means to prevent the model from putting too much emphasis on fluency during the fine-tuning stage at a cost of losing original intents. Thus, it pays to further leverage RL for the reformulator to adapt to reward signals from downstream environments like IC. This can correct the course from semantic drift, even though in the RL process, sequence fluency is impaired.

Discussions

We show that reinforcement learning objective can be used to train NLP reformulation models, which adapt to black-box systems like question answering and intent classification through reward signals. RL methods based on policy-gradient and self-critical training achieve better downstream performance compared to other alternatives. We find that while optimizing rewards from a QA system, reformulators struggle to maintain the ability to generate fluent questions compared to before RL. Aligned with recent progress in NLP, transfer learning from a pre-trained transformer has proven critical for a reformulation model to obtain implicit linguistic knowledge. The text-to-text framework provides flexibility for fine-tuning on paraphrasing and denoising datasets, as well as creating a model to provide numeric fluency scores for given textual sequences driven by human evaluations on the question well-formedness data. The fine-tuned T5 model is capable of generating quality out-of-sample reformulations before RL. This provides a more robust starting point than the previous LSTM translation-based model for later RL tuning. After RL on a specific black-box environment, this text-to-text reformulator demonstrates its ability to maintain fluency qualitatively and quantitatively while acquiring rewards from the downstream task. Further work will consider how to stabilize RL training for T5 models to avoid reward plateau in order to obtain a reformulator model that can continue to acquire rewards beyond a few epochs while maintaining fluency. Additionally, whether the T5 well-formedness model is biased towards the T5 reformulator models is in question, if present, a different type of transformer with better classification accuracy on QW may be employed to reduce such bias. Finally, as the text-to-text framework is more flexible in nature, swapping out the QA systems, systematically studying generalization on other out-of-domain datasets, and adding conversational context to produce more informed reformulations can be promising directions to further solidify the appeal of our approach.

Acknowledgement

The authors thank Scott Sanner for discussions and feedback, and DSTC-9 reviewers for helpful comments and suggestions.

Appendix - Qualitative Examples

Model	Query 1	Query 2
0. No Model (Original Query)	*1909 nobel prize winner failed entrance exams univ bologna , italy*	*2000 film jackie chan old west could funnier?"*
1. Pretrained (no RL) Google	How many won univ bologna 's italy nobel prize won?	Where is the chan old west west?
2. Tuned with RL (using PG) Google	What is is 1909 nobel prize winner failed entry exams univ bologna , italy name?	What is is is 2000 is is chan jackie chan old west might funnier name?
3. Policy gradient (PG) baseline	What is 1909 nobel prize winner failed entrance exams univ bologna , italy name name?	What is 2000 film jackie chan Old west might funnier name?
4. Self-critical training (SC)	Name of orange nobel prize winner exam bologna ' italy name 02 name univ bologna """"	Where is chan old west west west horse is located 2000 west west pogamerum is funnier name 2000 chan chan old west west name
5. PG + SC + fluency metric	Where is the name of the 1909 nobel prize win entrance exams univ bologna , italy?	What is the name jackie chan old west is funnier from 2000 chan old west west is chan old west is it made from the is it
6. T5 (Quora, MQR)	How to prove that a 1909 Nobel Prize winner failed entrance exams at the University of Bosnia and Herzegovina, Italy?	Can the 2000 film Jackie Chan's Old West be funnier?
7. T5 (Quora, MQR, RL)	What 1909 nobel prize winner failed entrance exams univ bologna, italy?	Is 2000 film Jackie chan old west could funnier?

Table 3: Comparisons on SearchQA Dataset between AQA and T5

Original Text	T5 Reformulations (no RL)	T5 Reformulations (RL)
How can we add new authorized people to an organization?	How do we create new authorized people within an organization?	How can we add new authorized people to an organization?
Why did we have to sell from client's ARF in order to cover these CWR checks?	Why did we have to sell from client's ARF in order to cover these CWR checks?	Why did we have to sell from client's ARF in order to cover these CWR checks?
Can you look up micro deposits for client?	Can you look up micro deposits for a client?	Can you look up micro deposits for clients?
unable to setup aes	How do I configure AES?	Why am I unable to setup aes?
Process for One time wire	What are the best methods for creating one time wire?	What is the process for a one time wire?
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?
wants to gift from the joint TA?	Can I gift from the joint TA?	Does one want to gift from the joint TA?

Table 4: T5 Sample Reformulations for Internal Query Dataset

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?

Table 5: Comparisons on Internal Query Dataset between AQA and T5

References

[2020] Angermüller, C.; Dohan, D.; Belanger, D.; Deshpande, R.; Murphy, K.; and Colwell, L. J. 2020. Model-based reinforcement learning for biological sequence design. In *ICLR*.

[2018a] Buck, C.; Bulian, J.; Ciaramita, M.; Gajewski, W.; Gesmundo, A.; Houlsby, N.; and Wang, W. 2018a. Analyzing language learned by an active question answering agent. *ArXiv* abs/1801.07537.

[2018b] Buck, C.; Bulian, J.; Ciaramita, M.; Gesmundo, A.; Houlsby, N.; Gajewski, W.; and Wang, W. 2018b. Ask the right questions: Active question reformulation with reinforcement learning. *ArXiv* abs/1705.07830.

[2006] Callison-Burch, C.; Osborne, M.; and Koehn, P. 2006. Re-evaluation the role of bleu in machine translation research. In *EACL*.

[2018] Cer, D. M.; Yang, Y.; yi Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.-H.; Strope, B.; and Kurzweil, R. 2018. Universal sentence encoder. *ArXiv* abs/1803.11175.

[2020] Chhina, N. S. 2020. Identifying well-formed questions using deep learning. *UVicSpace*.

[2020] Choshen, L.; Fox, L.; Aizenbud, Z.; and Abend, O. 2020. On the weaknesses of reinforcement learning for neural machine translation. *ArXiv* abs/1907.01752.

[2020] Chu, Z.; Chen, M.; Chen, J.; Wang, M.; Gimpel, K.; Faruqui, M.; and Si, X. 2020. How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions. *ArXiv* abs/1911.09247.

[2019] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

[2017] Dunn, M.; Sagun, L.; Higgins, M.; Güney, V. U.; Cirik, V.; and Cho, K. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv* abs/1704.05179.

[2019] Elgohary, A.; Peskov, D.; and Boyd-Graber, J. L. 2019. Can you unpack that? learning to rewrite questions-in-context. In *EMNLP/IJCNLP*.

[2013] Fader, A.; Zettlemoyer, L.; and Etzioni, O. 2013. Paraphrase-driven learning for open question answering. In *ACL*.

[2019] Falcon, W. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* 3.

[2018] Faruqui, M., and Das, D. 2018. Identifying well-formed natural language questions. In *EMNLP*.

[2013] Figueroa, A., and Neumann, G. 2013. Learning to rank effective paraphrases from query logs for community question answering. In *AAAI*.

[2018] Ge, T.; Wei, F.; and Zhou, M. 2018. Fluency boost learning and inference for neural grammatical error correction. In *ACL*.

[2020] Goutham, R. 2020. Paraphrase any question with t5 (text-to-text transfer transformer) - pretrained model and training script provided. *Medium. Note: https://towardsdatascience.com/paraphrase-any-question-with-t5-text-to-text-transfer-transformer-pretrained-model-and-cbb9e35f1555*.

[2019] Guo, Y.; Liao, Y.; Jiang, X.; Zhang, Q.; Zhang, Y.; and Liu, Q. 2019. Zero-shot paraphrase generation with multilingual language models. *ArXiv* abs/1911.03597.

[2018] Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2018. A deep generative framework for paraphrase generation. In *AAAI*.

[1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9:1735–1780.

[2017] Jaques, N.; Gu, S.; Turner, R.; and Eck, D. 2017. Tuning recurrent neural networks with reinforcement learning. *ArXiv* abs/1611.02796.

[2020] Keneshloo, Y.; Shi, T.; Ramakrishnan, N.; and Reddy, C. 2020. Deep reinforcement learning for sequence-to-sequence models. *IEEE Transactions on Neural Networks and Learning Systems* 31:2469–2489.

[2019] Keneshloo, Y.; Ramakrishnan, N.; and Reddy, C. 2019. Deep transfer reinforcement learning for text summarization. In *SDM*.

[2018] Kudo, T., and Richardson, J. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

[2020] Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv* abs/1909.11942.

[2020] Lin, S.-C.; Yang, J.-H.; Nogueira, R.; Tsai, M.-F.; Wang, C.-J.; and Lin, J. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *ArXiv* abs/2004.01909.

[2004] Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.

[2018] Liu, H.; Rong, W.; Shi, L.; Ouyang, Y.; and Xiong, Z. 2018. Question rewrite based dialogue response generation. In *ICONIP*.

[2019] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv* abs/1907.11692.

[2010] Madnani, N., and Dorr, B. J. 2010. The circle of meaning: from translation to paraphrasing and back.

[1979] McKeown, K. 1979. Paraphrasing using given and new information in a question-answer system. In *ACL*.

[1988] Meteer, M., and Shaked, V. 1988. Strategies for effective paraphrasing. In *COLING*.

[2013] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing atari with deep reinforcement learning. *ArXiv* abs/1312.5602.

[2017] Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural discrete representation learning. In *NIPS*.

[2002] Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

[2019] Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv* abs/1910.10683.

[2018] Rane, P. P.; Sargar, A.; and Shaikh, F. 2018. Self-critical sequence training for image captioning.

[2019] Roy, A., and Grangier, D. 2019. Unsupervised paraphrasing without translation. In *ACL*.

[2019] Ruder, S.; Peters, M. E.; Swayamdipta, S.; and Wolf, T. 2019. Transfer learning in natural language processing. In *NAACL-HLT*.

[2017] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *ArXiv* abs/1707.06347.

[2017] Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. *ArXiv* abs/1611.01603.

[2020] Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R. J.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2020. Learning to summarize from human feedback. *ArXiv* abs/2009.01325.

[2014] Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *ArXiv* abs/1409.3215.

[1999] Sutton, R.; McAllester, D. A.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

[2019] Syed, B.; Indurthi, V.; Gupta, M.; Shrivastava, M.; and Varma, V. 2019. Inductive transfer learning for detection of well-formed natural language search queries. In *ECIR*.

[2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *ArXiv* abs/1706.03762.

[2019] Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *ArXiv* abs/1905.00537.

[2020] Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; and Weston, J. 2020. Neural text generation with unlikelihood training. *ArXiv* abs/1908.04319.

[1991] Williams, R. J., and Peng, J. 1991. Function optimization using connectionist reinforcement learning algorithms. *Connection Science* 3:241–268.

[2019] Witteveen, S., and Andrews, M. 2019. Paraphrasing with large language models. *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

[2019] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv* abs/1910.03771.

[2016] Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, L.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G. S.; Hughes, M.; and Dean, J. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv* abs/1609.08144.

[2019a] Yang, Q.; Huo, Z.; Shen, D.; Cheng, Y.; Wang, W.; Wang, G.; and Carin, L. 2019a. An end-to-end generative architecture for paraphrase generation. In *EMNLP/IJCNLP*.

[2019b] Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

[2020] Yu, S.; Chen, Y.; and Zaidi, H. 2020. A financial service chatbot based on deep bidirectional transformers.

[2009] Zhao, S.; Lan, X.; Liu, T.; and Li, S. 2009. Application-driven statistical paraphrase generation. In *ACL/IJCNLP*.