Critical to Deep learning’s success in real-world applications is the ability to build models that generalize well to unseen data and can therefore cover multiple domains. In particular, in Question Answering (QA) tasks in NLP, models struggle to generalize and often overfit to specific datasets, making them unreliable for tasks on new datasets. It is therefore critical to develop methodologies to train domain-agnostic QA models. In other words, a model that learns domain invariant features to generalize to unseen data. We propose using an Adversarial Training framework, similarly to GANs in computer science (4). Our model has two main components: (1) the QA and (2) a domain discriminator. Finally, to further increase the robustness, Easy Data Augmentation (7) will be used to augment training data.

Adversarial QA

We implemented the Adversarial QA architecture proposed by (7) on a pretrained DistilBert encoder.

- Architecture: The architecture jointly trains a span classification head for the QA task and a discriminator head for the multi-class domain classification.
- Training: The training proceeds by iteratively training an augmented QA loss and a domain classifier loss.
- Discriminator Loss: Cross entropy loss $L_d$.
- Augmented QA Loss: Span classification loss $L_{sp}$ and a scaled KL divergence between the uniform distribution and the discriminator’s prediction $L_{ad}$.

$$L_{ad} = L_{sp} + \alpha L_{d}$$

(1)

Easy Data Augmentation

In order to generalize well, the model needs to learn how to deal with the data from a different domain. Given that the out of domain training set only contains a dozen samples, we suggest to apply easy augmentation techniques that will not perturb the general knowledge but will add more flexibility and make the distribution of the data less domain specific. To that extent, we mainly work with basic data augmentations (7):

1. Synonym replacement: Every word is replaced by its synonym with a probability $p_{syn}$.
2. Random Deletion: Every word is deleted with probability $p_{del}$.

Both of the augmenting techniques mentioned above, are applied only to the part of the context that does not contain the answer. The data is augmented $X$ times.

In-domain training and validation

- Baseline: pretrained DistilBert model with a Question Answering head fine-tuned to the in-domain data.
- Adversarial: pretrained DistilBert model with the adversarial architecture fine-tuned to the in-domain data.

Few-shot out-of-domain train and validation

- Baseline: Fine-tuned: Baseline model trained on small train set of out-of-domain data.
- Adversarial: Fine-tuned + Freeze DistilBert: Adversarial model trained on small train set of out-of-domain data while freezing the DistilBert parameters.

Training Loss for In-domain data

The following plot shows the Baseline and Adversarial training. We can see that the Adversarial architecture is doing a good job at "controlling" the discriminator while still reinforcing the QA loss.