Debiasing Models, Dataset Augmentation and MoE for Out-of-domain Generalization

Stanford CS224N {Default} Project
Track: {RobustQA}

Akash Velu
Department of Computer Science
Stanford University
avelu@stanford.edu

Manasi Sharma
Department of Computer Science
Stanford University
manasis@stanford.edu

Nishant Jannu
Department of Mechanical Engineering
Stanford University
nishant3@stanford.edu

Abstract

Question answering (QA) has gained wide significance in recent years with the rise of chatbots and more powerful search engines, but the performance of current models is still quite limited in *domain adaptation* settings in which models which are trained on source training domains are tasked with generalizing to new target data distributions where training data is limited. In this work, we hypothesize that in QA settings, models fail to generalize to new data domains because they have overfit to spurious correlations in the source domain training sets which do not generalize to the target data distributions. Motivated by this hypothesis, we implement several approaches to prevent QA models from overfitting to spurious correlations – dataset augmentation with synonym replacement, debiasing techniques to prevent models from using biased features during training, and lastly, ensembling methods which combine multiple trained models with a mixture of experts. Early results show that the synonym augmentation approach improves upon the several baselines, whereas debiasing approaches does not achieve improved performance. We explore several reasons for these results and discuss future steps to improve performance in domain generalization for question answering.

1 Key Information to include

- Mentor: Sarthak Kanodia
- External Collaborators: No
- Sharing project: No

2 Introduction

Question answering (QA) is a prevalent and well-researched problem in NLP, with applications in tasks such as information retrieval and the development of products such as dialog systems and chatbots. However, while NLP models have achieved results superior to those of humans in various such datasets, these models often fail to generalize beyond this dataset: past works [1] have shown that models overfit to the source dataset on which they were trained, and fail to adapt to new domains without further training. One potential reason for this patter is that models learn source-domain specific biases such as predicting based on text near question-words as answers (regardless of context)

(i.e., lexical overlap) or guessing an answer based on how early it occurs in the context (i.e., position bias), and fail to capture patterns which are important when generalizing to new domains. As a result, models that encode these biases are brittle and show poor performance on data that is dissimilar to in-domain data (e.g. data with low lexical overlap).

We noticed this when looking more closely at the indomain vs. out-of-domain datasets. In the following example from SQuAD, we can see that there is a very high overlap between the words in the question and the words leading up to the answer in the context (called "lexical overlap"). Such a paradigm can make it very easy for a model to expect the answer to lie in the sentence with the most words in common with the question.

Question: *"How many incoming students did Notre Dame* ***admit in fall 2015****?"*
Context: *"Notre Dame is known for its competitive admissions, with the incoming class enrolling in* ***fall 2015 admitting 3,577*** *from a pool of 18,156 (19.7%). The academic profile of the enrolled class continues to rate among the top 10 to 15 in the nation for national research universities..."*
Answer: *"3,577"*

On the other hand, as we can see from this example from the RACE dataset, the answer is much more complicated to grasp, as it requires higher-order understanding of the relationship between multiple sentences (connecting high shutter speed being "helpful" for any wildlife to lion being an example of active wildlife). This cannot be gleaned from searching for high lexical overlap, as a model trained on the indomain data is likely to do.

Question: *"What will contribute to a satisfactory photo of a* ***running lion*** *in the wild?"*
Context: *"...Zoom and shutter speed For action or crowd shots, a fast shutter speed is a key factor. ¨When dealing with anything that's active–****wildlife*** *or people in action on the street–faces change within a tiny part of a second,¨said Arnold,* ***¨a fast shutter speed is helpful*** *in shooting the several hundred photos you might need to get that single winning shot.¨"*
Answer: *"shutter speed"*

Enabling QA models to generalize beyond their training dataset is an important task: training new large-scale models for each domain can become prohibitively expensive and inefficient. Training models which can perform strongly on the training dataset, as well as generalize to other domains by utilizing properties of the training set, is therefore an active area of research known as *domain generalization*. Past works have sought to tackle the domain generalization challenge with large language models such as BERT [2] which are pre-trained on large text corpora and fine-tuned on the available training data from both the source data domains and target data domains. However, BERT and other large language models are still negatively impacted by dataset biases and spurious correlations in training data.

In this work, we aim to make large language models (LLMs) such as BERT more robust to dataset biases present in QA settings. Our methods are motivated by the hypothesis that LLMs overfit to dataset biases in the source training data – more specifically, lexical overlap biases in which the question and answer contain high overlap in the words that are present. To prevent our models from fitting to these biases, we explore the following three complementary approaches:

1. **Dataset Augmentation with Synonym Replacement**: to add data which does not contain high lexical overlap between question and answer, we replace a specified fraction of words in the question with synonyms from a WordNet model [3], and train a DistilBERT [4] model on this augmented dataset.

2. **Debiasing models**: inspired by [5], we train a simpler, *biased* model with TF-IDF [6] features alongside a DistilBERT model which is encouraged to learn features not captured by the biased model.

3. **Ensembling with mixture of experts**: to combine the outputs of an ensemble of high-variance predictors, we implement a mixture-of-experts model which outputs a weighted combination of the predictions of the ensemble. We explore several ensembling techniques in which the ensemble models are trained with debiasing (approach 2) and are trained on various source domain sets.

Our code can be found on Github [1].

3 Related Work

Domain generalization has been a well-studied topic in a variety of different fields. within NLP, several popular approaches have focused on addveersarial training approaches in which models are trained to learn *domain-invariant* features which are able to generalize to new domains [7]. These approaches have also shown promise in other fields such as computer vision [7].

Other works have adopted dataset augmentation approaches which either increase the volume of data in the target domains or improve the quality of source / target domain data through methods such as synonym replacement and backtranslation [8]. Other methods have proposed analyzing the dataset itself when making a prediction, for instance, by finding the most correlated in-domain dataset compared to the training sample and using the model associated with that dataset for predictions. Other works [9] have adopted meta-learning approaches to adapt in few-shot ways to new data domains. In this work, we adopt ideas from ensembling via mixture of experts [5] and dataset augmentation and apply them to QA settings.

4 Approach

In this section, we outline the three approaches discussed in Sec. 3 in more detail. To emphasize, these approaches are motivated by the core hypothesis that the source domain sets have questions which are much easier to answer than those in the test domain sets, particularly due to high lexical overlap between the questions and answers in the source data domains. We describe each approach individually before discussing how each method connects with one another. For each method, we assume access to a set of a *source* in-domain training datasets and a set of *target* out of domain training sets.

4.1 Dataset Augmentation with Synonym Replacement

To both increase the quantity of source-domain training data and to improve the quality of source-domain training data, for each source domain training point, replace *cp%* of the words in the question with synonyms obtained from a pre-trained WordNet [3] model. We then used this augmented source-domain dataset to fine-tune a pre-trained DistilBERT model, which is subsequently fine-tuned on the target-domain training sets.

Specifically, our synonym replacement approach involves the following steps to ensure high quality of the synonyms which are chosen to replacement words in the sentence:

1. Isolate the words that were in common with the context and the question (called "lexical overlap")
2. POS-tag every word in the sentence using WordNet's dependency parser nltk.pos_tag (which is trained on the Treebank corpus) to determine which words to replace with synonyms – we ensure that the words that are replaced are non-proper nouns, verbs, adjectives, or adverbs and do not replace stop-words. We also utilize stemming to prevent synonyms from being morphological variants of the words they replace, and dependency parsing to ensure that the synonym is the same part of speech of the word it replaces.
3. Iterate through each word in the question; after checking it's lexical overlap performing a tagging check, replace the word with a synonym with a probability of *cp*.

4.2 Debiased Models

The core idea behind the synonym replacement approach was to augment the *data* with samples that would prevent the model from overfitting on spurious biases in the source domain data. In this method, to more expicitly enable the DistilBERT model to learn patterns which are *not* biased or based on lexical overlap and instead focus on aspects of the data which are more generalizable, we implement debiasing approaches described in [5]. In this approach, a "biased" model is first trained

[1] https://github.com/akashvelu/cs224n_project

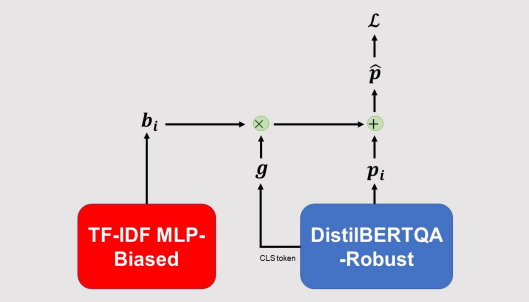

Figure 1: Debiasing model architecture.

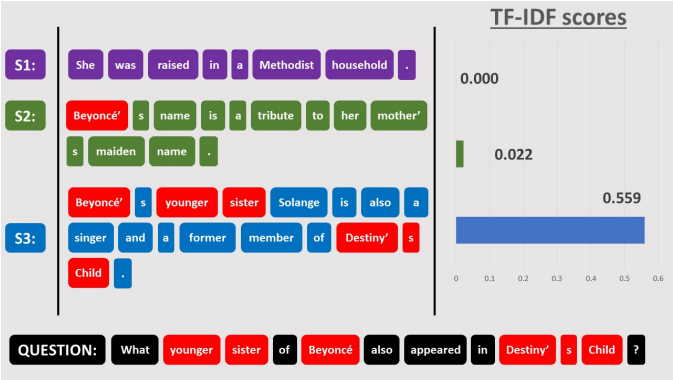

Figure 2: TF-IDF scores example.

on the in-domain source datasets, and is set-up such that it learns based on superficial patterns in the data (e.g. lexical overlap with the question words). Using this biased model (which is now frozen), a "robust" model is then trained in a manner such that it is encouraged to learn other non-trivial patterns in the data that are more likely to generalize. Specifically, predictions of the biased model (denoted as b) and the robust model (denoted as p) are combined to create a final prediction $\hat{p}$ in the following manner:

$$\hat{p} = softmax(\log(p) + g(x)\log(b))$$

Here, g is a learned *mixing* function that determines how much to weight the prediction of the bias model for an input x. Note that during backpropagation when training the robust model, the weights of the biased model are not updated. The robust model is subsequently fine-tuned on the out of domain datasets. We call this method with a learned function g "Debiased Learned-MixIn", and without the learned function "Debiased".

To ensure that the biased model learns based on biases in the source-domain datasets, its input is computed with TFIDF scores which capture the overlap between the question and various sentences in the context. The architecture of the biased model is additionally limited to a multi-layer perceptron (MLP) to reduce its representational power compared to the robust model. The weighting model g is learned MLP which takes as input the CLS token representation (output by a pre-trained DistilBERT model) of the context and question.

As detailed by [5], to prevent the mixing model g from assigning a weight of 0 to the biased model and ignoring the predictions of the biased model, we also implement a variant of this approach in which the following entropy penalty term R is added to the standard-cross entropy loss:

$$R = wH(softmax(g(x)log(b))$$

Here, w is a hyperparameter which determines the weight given to the entropy term in the loss. To remain consistent with [5], we call debiasing with this entropy penalty "Debiased Learned-Mixin + H".

For the debiasing methods, we investigated the performance of debiased models trained on all source datasets and fine-tuned on all target datasets, as well as the performance of an ensemble of debiased models which were trained on individual source domain datasets on fine-tuned on individual target domain datasets.

4.3 Mixture of Experts

Lastly, to take advantage of multiple high-variance models which were trained on the QA task at hand, we implemented a mixture-of-experts (MoE) [10] based ensembling method which combines the predictions of an ensemble of pre-trained predictors $F = \{f_1, ..., f_k\}$ through a weighted sum:

$$f(x) = \sum_{i=1}^{k} w_i(x) f_i(x)$$

Here, w is a learned "mixer" function which determines how much weight to give to each "expert" predictor in the ensemble. The primary motivation for implementing this method arose from the fact that the debiased models often performed well when trained on a singular in-domain source dataset and fine-tuned on a singular out-of-domain target dataset, but performed worse when trained and fine-tuned on all datasets at once. We hypothesized that combining the predictions of multiple debiased models which were individually trained on various individual in-domain source datasets and fine-tuned on different individual out-of-domain target datasets would result in improved overall performance.

Motivated by this observation, we learn a separate ensemble F_j and mixer w_j for each out-of-domain target dataset j. To make predictions at evaluation time, we implement two methods. In the first, for each evaluation datapoint, we assume access to knowledge of the out-of-domain dataset that the datapoint comes from and choose the corresponding ensemble and mixer. We call this method "MoE". In the second, to avoid the assumption of the knowledge of the out-of-domain dataset for each datapoint, we first train a DistilBERT classifier to predict which out-of-domain target dataset a given datapoint comes from, and use this prediction to then choose the corresponding ensemble and mixer model. We call this method "MoE+Classifier".

In our implementation w is a learned MLP which takes as input the CLS token representation (from a pre-trained DistilBERT model) of the context and question. We additionally experiment with various ensembles – in particular, F_j is either a set of *debiased* models (trained with the technique described on 4.2) which were first trained *separately* on the (three) in-domain source datasets (resulting in three models) and then individually fine-tuned on the $j - th$ out-of-domain dataset, or a set of non-debiased models which again were were first trained separately on the (three) in-domain source datasets (resulting in three models) and then individually fine-tuned on the $j - th$ out-of-domain dataset.

- **Out-of-domain finetuning:** First, we trained the provided DistilBERTQA model on the source domain datasets, and then subsequently fine-tuned this model on the small out-of-domain training sets.

5 Experiments

5.1 Data

We use the datasets specified in the default final project guide. Models are typically first trained on a combination of the in-domain train sets (Natural Questions, NewsQA, and SQuaD) and are fine-tuned on a combination of the out-of-domain train sets (RelationExtraction, DuoRC, and RACE. The validation data is a combination of the in-domain of out-of-domain datasets.

5.2 Evaluation method

We will be using the Exact Match (EM) score and F1 score metrics (as defined in the default project handout).

5.3 Experimental details

All experiments are performed under a batch size of 16 and a learning rate of 3e-5. Source domain training is performed for 3 epochs, and models are evaluated on OOD validation sets every 10 batches, during which the best model weights are recorded. We perform limited grid-search hyper-parameter sweeps over learning rate (over the values 3e-05, 1e-05, and 1e-4) and and batch-size (over the values 16, 32).

Synonym Replacement: For the synonym replacement approach, *augment* the *source domain datasets* by replacing cp% of words in the question with synonyms obtained from WordNet. Specifically, we tried replacing all relevant words in the in-domain source question with synonyms (cp = 1.0), and tried replacing 50% of relevant words in the question with synonyms (cp=0.50). Synonym replacement was not used to augment the out of domain training set.

Debiasing models: We ran a series of experiments with the debiasing model approach. We first experimented with the architecture of the biased model. Wee first maintained the bias model to be a DistilBERT model trained in the in-domain source datasets, and operated on the assumption that this baseline model would overfit on features such as lexical overlap. The robust model was kept to be the same architecture and was trained via the scheme decribed in section 4.2.

The second approach follows [11], who suggest that the debiasing approach can fail if the "biased model" learns more than the features you want it to capture. To address this, we sought to use a much simpler model based on TF-IDF scores that only captured lexical overlap. TF-IDF scores for each indomain train sample were generated by utilizing the *TfidfVectorizer* [2] and *cosine similarity* [3] functions from Scikit-learn. A feature vector of the TFIDF scores were then fed into an MLP of one linear layer of size 384; this MLP was trained on the source domain datasets (with TFIDF features) for 20 epochs, with a learning rate of 1e-3. Again, the robust model was designed to be a DistilBERT model which was trained as is described in section 4.2, and is subsequently fine-tuned on the target domain train datasets.

MoE Ensembling with Debiasing

When training on all 3 source datasets and evaluating on all 3 target domain datasets, experiments with the debiasing approach depicted sub-optimal performance. Subsequently, we shifted towards training a robust model on a *single* source domain dataset, and fine-tuning it on a *single* target domain dataset. As there are three source datasets and three target domain datasets, this resulted in a total of nine models. Analysis of these models demonstrated that the debiased models often displayed improved performance (compared to a non-debiased baseline) in the target domain in which they were fine-tuned. This motivated the approach described in section 4.3.

For each robust model trained, the corresponding biased model was kept to be an MLP architecture with TFIDF inputs as described in section 5.3, whereas the robust models were DistilBERT models. The classifier model and mixer models w_j were also DistilBERT models with a linear layer appended to the CLS token output. The classifier model is trained on the target domain data and achieved an accuracy of 93% in validation data.

During evaluation time, for a given sample, the classifier first predictions with target domain dataset the datapoint is from. The corresponding ensemble models F_j and the mixer model w_j are chosen, and are used to make the final prediction. We experiment with two prediction strategies: (a) a weighted sum in which the final prediction is a weighted linear sum of the ensemble predictions (with weights determined by the mixer) and (b) a pure prediction strategy in which the prediction of the ensemble model with the highest weight is used (caalled choose

5.4 Results

The best synonym replacement model improved upon the baseline F1 score by 2.45 %. However, neither of the debiased or MoE models resulted in better scores.

We noticed that the debiased models trained separately on each of the in-domain datasets perform better than their non-debiased equivalents. However, training on all the source datasets together-results in a degraded performance. On breaking up the results by out-of-domain dataset, we found

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

Training Run	EM-Scores	F1-Scores
Baseline	34.55	49.88
Baseline Fine-tuned	36.13	50.24
Synonym Replacement (cp = 1.0)	34.03	48.78
Synonym Replacement (cp = 0.5)	**36.91**	**51.10**
Debiased Model	32.46	49.07
Debiased Model (with entropy penalty)	32.72	48.83
MoE with non-debiased models	34.03	48.34
MoE with debiased models	34.35	49.48

Table 1: Final results for the experiments, on the dev validation set. Baseline denotes a DistilBERT model trained only on source domain datasets. Baseline Fine-tuned denotes a DistilBERT model trained in source domains and fine-tuned on target domains. The other rows indicate results from the methods described in section 4, with each model being trained on all source domains and fine-tuned on all target domains.

that the MoE model is better than the 3 relevant expert models in the RACE DuoRC datasets, but not the Relation Extraction dataset. We also saw that using the weighted result instead of sampling the model with the highest weight gave better scores.

As shown in the Appendix, we also noticed that ensembling a set of models with a MoE often improves upon the performance of any given model in the ensemble; this is particularly true in the RACE and DuoRC datasets, in which ensembling improved upon the performance of any individual model (this is not true for Relation Extraction).

Our highest performing test set scores were from the Synonym Replacement (cp = 0.5) with scores F1: 58.564 and EM: 40.986, and achieved a EM score rank of 6 and F1 score rank of 21 in the validation leaderboard (as of March 14th).

6 Analysis

Error Analysis for Debiased Model

We looked at the predictions made by the debiased model vs our best model (synonym replacement) to find insights on how the debiased model was performing- in terms of how it was making predictions differently, what it got right and where it went wrong. In Table 2, we consider two examples from the RACE dataset- one in which the debiased model predicts the correct span and one in which it does not.

In Example 1, the word 'computers' (key word that appears in the question) does not occur in the sentence containing the answer and the debiased model is able to use other strategies to identify the correct answer span while the best model suffers. However, in the Example 2, the answer is contained in a sentence with semantically similar words as the question ('poor'/'depressed', 'subjects'). The biased model identifies a different sentence in this case as well.

From these examples, it is clear that the biased model captures alternative strategies to lexical overlap which helps it answer questions where the answer is contained in regions of lower lexical overlap with the question . This is demonstrated in other examples we evaluated as well. However, a major drawback of the debiasing seems to be that it forces the model to look away from the high-overlap region even in examples where the answer span is contained in this region. We wanted the model to selectively look elsewhere when it recognises that the biased model prediction is unreliable. However, it seems to do it almost always- indicating that the weighting function g doesn't work as expected.

Validation of Hypothesis on Different Source Datasets

Another reason for the debiasing method performing below our expectations can be attributed to the fact that the lexical overlap bias is less dominant in the News QA and Natural Questions datasets in comparison to SQuAD. Table 3 shows the improvement in F1 scores made by models (baseline and debiased) trained separately on the the three source datasets when evaluated on the OOD Validation datasets. Clearly, SQuAD gains significantly in comparison to the other two. Other biases such as positional bias (Ex: preference for choosing answers that occur near the start of the context) might have been more applicable choices for them.

	Example 1	Example 2
Context	In industry, computers mean automation , and automation means unemployment. Computers in the United States have already begun to take the place of workers whose tasks are simple. The variety of jobs, done only by humans in the past, which the machine can perform more rapidly, accurately and economically, increases with each new generation of computers. If we follow this trend, we will be faced with mass unemployment for all but a handful of highly trained professionals who will be more powerful and overworked than they are now.	The teaching arrangement filled me with fear. I was to divide the class of twenty-four boys, aged from seven to thirteen, into three groups and teach them all subjects–including art, football, cricket and so on–in turn at three different levels. Actually, I was depressed at the thought of teaching algebra and geometry–two subjects in which I had been rather weak at school.
Question	Which kind of the following persons will be the first to be employed if computers continue to develop?	Which subjects was the writer poor at?
Label	highly trained professionals	algebra and geometry
Debiased Model Prediction	highly trained professionals	art, football, cricket
Best Model Prediction	workers whose tasks are simple	algebra and geometry

Table 2: Comparison of debiased model predictions on different types of examples. Debiased Model Prediction refers to the prediction of a debiased model trained with the entropy loss term, and best model prediction refers to the prediction of the best trained model, which uses a synonym replacement dataset augmentation technique.

Model	SQuAD	News QA	Natural Questions
Baseline	42.61	40.55	37.90
Debiased	44.53	41.02	38.11
% Improvement	4.5	1.2	0.55

Table 3: F1 scores on OOD Validation Set. Baseline refers to to a DistilBERT model trained only on source domains. Debiased refers to a DistilBERT model trained only in the source domains with the debiasing scheme discussed in Section 4

Model performances on Different OOD Datasets: All the models considered show especially poor performance on the RACE dataset (Refer tables in Appendix). RACE is created by teachers for Chinese School English Examinations while the other datasets are crowd-sourced. Thus, the questions on it require higher ordering reasoning to answer and all the methods we implemented motivate the model to address this challenge.

7 Conclusion

Our core hypothesis that source domain biases prevent models from generalizing to target domains was validated by the improved performance of the synonym replacement method. Tackling this challenge with debiasing models demonstrated promising but overall worse results, suggesting that further investigations into this technique and its failure points would be an interesting direction of investigation. **What we learned:** critically analyzing past papers and works, implementing methods we thought were promising, and developing our own extensions was a fun and educative process. **Limitations of our work:** Our work does not extensively investigate reasons for the suboptimality of the debiased model approaches past limited qualitative analysis. We also did not extensively tune the weighting model in the debiasing approach; performing better hyperparameter sweeps would be a good next step. For future work, we will look at methods to automatically detect source dataset biases.

References

[1] Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. Learning and evaluating general linguistic intelligence, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Mike Wallace. *Jawbone Java WordNet API*, 2007.

[4] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[5] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *CoRR*, abs/1909.03683, 2019.

[6] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[7] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[8] Sravana Reddy and Kevin Knight. Obfuscating gender in social media writing. In *Proc. of Workshop on Natural Language Processing and Computational Social Science*, 2016.

[9] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[10] Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. Adaptive mixture of local expert. *Neural Computation*, 3:78–88, 02 1991.

[11] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Learning to model and ignore dataset bias with mixed capacity ensembles. *CoRR*, abs/2011.03856, 2020.

A Appendix

We include the complete set of results for all the experiments we ran below, divided up by each of the out-of-domain datasets. Here we compare the performance between the methods we ran (as opposed to comparison with the baseline performance).

Debiasing and MoE Ensembling approaches:

- **Results on DuoRC dev set:**

Training Run	**EM-Scores**	**F1-Scores**
Debiased trained on NewsQA only	28.57	40.57
Debiased trained on Natural Questions only	21.43	33.35
Debiased trained on SQUaD only	23.81	32.55
MoE Ensembled debiased model (weighted sum)	**32.54**	**41.65**
MoE Ensembled debiased model (choose highest weight)	21.43	33.35

- **Results on RACE dev set:**

Training Run	EM-Scores	F1-Scores
Debiased trained on NewsQA only	17.97	30.37
Debiased trained on Natural Questions only	11.72	23.80
Debiased trained on SQUaD only	14.84	30.29
MoE Ensembled debiased model (weighted sum)	**17.97**	**32.88**
MoE Ensemble debiased model (choose highest weight)	14.84	29.54

- **Results on RelationExtraction dev set:**

Training Run	EM-Scores	F1-Scores
Debiased trained on NewsQA only	**55.47**	**75.11**
Debiased trained on Natural Questions only	54.69	73.15
Debiased trained on SQUaD only	53.12	71.90
MoE Ensembled debiased model (weighted sum)	54.69	73.65
MoE Ensemble debiased model (choose highest weight)	54.69	71.69

- **Summary of important results:**

Training Run	DuoRC		RACE		RelationExtraction	
	EM-Scores	**F1-Scores**	**EM-Scores**	**F1-Scores**	**EM-Scores**	**F1-Scores**
Debiased model trained on NewsQA only	28.57	40.57	17.97	30.37	**55.47**	**75.11**
Debiased model trained on Natural Questions only	21.43	33.35	11.72	23.80	54.69	73.15
Debiased model trained on SQUaD only	23.81	32.55	14.84	30.29	53.12	71.90
MoE Ensembled debiased model (weighted sum)	**32.54**	**41.65**	**17.97**	**32.88**	54.69	73.65
MoE Ensembled debiased model (choose highest weight)	21.43	33.35	14.84	29.54	54.69	71.69