An Analysis on the Effect of Domain Representations in Question Answering Models

Stanford CS224N Default Project (Robust QA track)

Nick Vasko
Department of Computer Science
Stanford University
nvasko@stanford.edu

Abstract

Studies of robust reading comprehension models have included both learning domain specific representations and domain invariant representations. This project analyzes the effectiveness of each of these approaches using Mixture-of-Experts (MoE) and adversarial models. In the domain specific approach, MoE's form a single expert model for each input domain (Guo et al. [1], Takahashi et al. [2]). In contrast, domain invariant models learn a generalized hidden representation that cannot distinguish the domain of the input (Ma et al. [3], Lee et al. [4]). Additionally, models are assessed to determine their level of understanding of natural language against learning simple linguistic bias heuristics.

1 Introduction

Reading comprehension is a sufficiently difficult task that can assess the ability of machines to understand natural language. Recent progress has focused on using large pretrained models to achieve state-of-the-art results on the Machine Reading for Question Answering (MRQA) shared task (Takahashi et al., 2019 [2]; Lee et al., 2019 [4]). This task requires models to learn to generalize question answering to domains not seen during training. Although progress has been made, challenges remain in assessing whether these models can truly understand natural language (Sugawara et al., 2020) [5].

Common approaches explored in recent literature include: mixture-of-experts (MoE) models [2] and domain adversarial models [4]. These models differ in theory as MoE attempts to learn domain specific expert networks, while adversarial models attempt to learn domain invariant representations that can generalize well.

An additional area of research that motivates this work is simple bias heuristics for QA problems. Building debiased models is particularly interesting in assessing how well models truly understand natural language, rather than relying on simple linguistic patterns.

The focus of this project is to first implement an array of mixture-of-expert and adversarial models to compare their performance and representation of different domains. The second part of this work analyzes the amount of linguistic bias incorporated into these models to assess understanding of natural language. In order to do this, simple heuristics are studied to identify what types of question answering examples the models perform best on. The results of this work show that the models analyzed achieve much higher results on datasets that contain examples of simple heuristics. Additionally, it is shown that understanding the types and difficulty of questions from each dataset may provide motivation to explore additional work in student-teacher models that attempt to learn debiased representation similar to Wu et al., 2020 [6].

2 Related Work

Mixture-of-Experts. Guo et al., 2018 [1] and Takahashi et al., 2019 [2] have implemented mixture-of-experts (MoE) models for NLP tasks in a multi-domain setting. MoE models are composed of different neurel networks (experts), where each expert attempts to learn to handle a specific subtask, in this case a single domain. In [1], each expert is trained on an individual domain dataset among all the training datasets. The approach in [2] differs slightly in that a regularization loss term is used to control the amount of weight put on an individual expert.

Adversarial Models Adversarial models are trained using two components and have been applied to the QA setting in Lee et al., 2019 [4]. The idea is to train both a QA model and a discriminator model. The discriminator is trained on a specific subtask, while the QA model is trained to attempt to confuse the discriminator model. In this setting, the discriminator classifies the input into the domain it belongs to. Since the QA model learns to confuse the discriminator, the model is able to learn domain invariant features, while still predicting the correct answer.

Bias Patterns in NLP. Recent literature has studied whether large pretrained language models truly understand natural language. McCoy et al., 2019 [7] show that models learn simple syntactic structures from training data in natural language inference models. These models fail to generalize to more challenging tasks. Debiasing methods have been studied by Utama et al., 2020 [8] and Wu et al., 2020 [6] to attempt to create models that are adversarial to simple syntactic heuristics. These models attempt to learn deep understanding of knowledge that can generalize outside of the training domain.

3 Approach

In order to sufficiently analyze different domain representations and biases, several models are trained to determine if there are differences amongst architectures. In this section, the details of theses models are explained along with the biases that are studied.

3.1 Baseline Model

The baseline model is HuggingFace's DistilBERT Question Answering implementation. Cross entropy is used as the loss function, and will be referred to as $\mathcal{L}_{RC}$.

3.2 Mixture-of-Experts Models

MoE models consist K neural networks (experts) that learn to handle different in-domain datasets, and a gating network, which classifies the input representation to the correct domain expert. For out-of-domain generalization, the MoE computes a weighted average of the experts [2]. Here, the MoE takes a DistilBERT hidden representation, $H \in \mathbb{R}^{dxL}$, as input and outputs $Y \in \mathbb{R}^{dxL}$. Then, Y is fed into the final output layer. The MoE is defined with individual expert weights $\mathbf{W_i}$ and bias $\mathbf{b_i}$ as follows:

$$Y = \sum_{i=1}^{K} G(H)_i E_i(H) \tag{1}$$

$$E_i(H) = \mathbf{W_i} H + \mathbf{b_i} \tag{2}$$

3.2.1 MoE-Base

The MoE-Base model implements the structure of [2], which uses a bidirectional gated recurrent unit and a softmax linear layer as the gating network, with weights $\mathbf{W_g}$ and bias $\mathbf{b_g}$:

$$G(H) = softmax(\mathbf{W_g}[\overrightarrow{h_L}, \overleftarrow{h_1}] + \mathbf{b_g}) \tag{3}$$

$$\overrightarrow{h_L} = \overrightarrow{GRU}(H); \overleftarrow{h_1} = \overleftarrow{GRU}(H) \tag{4}$$

The loss function uses the reading comprehension cross-entropy and adds an importance loss term to control the variation in probabilities assigned to each expert. For simplicity, the cross task of natural

language inference is dropped from [2], making the loss:

$$\mathcal{L}_{MoE-Base} = \mathcal{L}_{RC} + \lambda_{importance} CV(\sum_{z \in Z} G(Z))^2 \tag{5}$$

where Z represents all the samples of a minibatch, $CV(\cdot)$ is the coefficient of variation, and $\lambda_{importance}$ is a hyperparameter. The CV loss term ensures a non-negligible probability is assigned to each expert.

3.2.2 MoE-Domain Classifier

The next implementation analyzed ensures each expert E_i is aligned to dataset D_i, for all $i \in \{1, ..., K\}$. The loss function adds a domain classification cross task to penalize when the gating network classifies the domain incorrectly. Motivated by [4], in this model the GRU gating network is replaced by a linear layer and only h_0, the hidden representation of the $[CLS]$ input token, is used as input. The gating network is defined as:

$$G(h_0) = softmax(\mathbf{W_g} h' + \mathbf{b_g}) \tag{6}$$

$$h' = ReLU(\mathbf{W_1} h_0 + \mathbf{b_1}) \tag{7}$$

The loss function with an added cross entropy loss is:

$$\mathcal{L}_{MoE-Domain} = \mathcal{L}_{RC} + CE(G(h_0)) \tag{8}$$

3.3 Adversarial Models

3.3.1 Adversarial-Baseline

The baseline adversarial model reflects the architecture of Lee et al., 2019 [4] [1]. The model has two components, a QA model and a discriminator. The difference here is that the model is trained to learn invariant domain representations.

The discriminator is trained to identify the domain of the input into one of K domains. The gating network from Section 3.2.2 is used for this. However, in this model, while training the discriminator only the gating network parameters are updated.

The QA model reflects the architecture of the baseline model in Section 3.1. An additional loss term, the Kullback-Leibler (KL) divergence, is added to confuse the discriminator during training. The goal is to minimize KL divergence such that the probabilities output by the discriminator do not differ from the uniform distribution across K domains.

The adversarial model is trained in two-steps for each minibatch. A gradient step is taken for the QA model, followed by a gradient step on the discriminator model.

3.3.2 MoE-Adversarial

The MoE-Adversarial model presents a novel approach to training adversarial and mixture-of-experts. Here a gating network from Equation 6 is used instead of a disciminator network. The gating network is trained to minimize the Kullback-Leibler (KL) divergence between uniform distribution over K classes denoted as U(l) and gating network's domain prediction. This archictecture is simpler than a the baseline adversarial network, as a second gradient step is not needed. In this setting, the MoE model is used as an ensemble to learn different features of the hidden state, rather than a domain classifier.

$$\mathcal{L}_{MoE-Adv} = \mathcal{L}_{RC} + \lambda_{adv} \mathcal{L}_{Adv} + \lambda_{importance} CE(G(h_0)) \tag{9}$$

The details of $\mathcal{L}_{Adv}$ are provided in Lee et al. (2019) [4]. The importance loss is added here to ensure that the gating network assigns uniform probabilities rather than emphasizing one expert.

3.4 Bias Methods

Wu et al., 2020 [6] implement a model that incorporates four debiasing methods. In this work, three of these biases are analyzed to determine if the above model architectures have learned to exploit these biases, rather than truly understand natural language concepts.

[1] Code adapted from `https://github.com/seanie12/mrqa`

Lexical Overlap. Lexical overlap occurs when the answer to a QA example is contained within the context sentence most similar to the question. To assess this concept, each context sentence is converted to an embedding space using Sentence-BERT [2]. Next, the cosine similarity is taken with respect to the question. The similarities are ranked and only the most similar sentence is kept and used during as the context.

Context Sentence 1: (Similarity: 0.36)
In the film Knute Rockne, All American, Knute Rockne (played by Pat OḂrien) delivers the famous "Win one for the Gipper" speech, at which point the background music swells with the "Notre Dame Victory March".
Context Sentence 2: (Similarity: 0.85)
George Gipp was played by Ronald Reagan, whose nickname **"The Gipper"** was derived from this role.
Context Sentence 3: (Similarity: 0.37)
This scene was parodied in the movie Airplane! with the same background music, only this time honoring George Zipp, one of Ted Strikerś former comrades.
Context Sentence 4: (Similarity: 0.27)
The song also was prominent in the movie Rudy, with Sean Astin as Daniel "Rudy" Ruettiger, who harbored dreams of playing football at the University of Notre Dame despite significant obstacles.
Question: Ronald Reagan had a nickname, what was it?

Figure 1: Example of Lexical Context from SQuAD. Cosine similarities are provided for each sentence. Context sentence 2 is most similar to the question, and is the only part kept in the bias example.

Interrogative Adverb Questions. *Wh-word* bias questions were studied by Weissenborn et al., 2017 [9]. These questions are identified and all words except the interrogative adverb is removed from the question. In these examples, the model must now only use this subset of the question to determine the answer.

Context: On **February 6, 2016**, one day before her performance at the Super Bowl, Beyoncé released a new single exclusively on music streaming service Tidal called "Formation".
Question: When ~~did Beyoncé release Formation?~~

Figure 2: Example of Interrogative Adverb Question from SQuAD. The model is only needs to use the word "When" from the question to predict the only date in the context.

Empty Question. The answer can be predicted correctly without the presence of the question. This bias tests to see if the model selects the most prominent entity of the context.

Context: Chopin's life was covered in a BBC TV documentary Chopin – The Women Behind The Music (2010), and in a 2010 documentary realised by ***Angelo Bozzolini and Roberto Prosseda*** for Italian television.
Question: ~~What are the names of the two people that created a documentary for Italian television?~~

Figure 3: Example of Empty Question from SQuAD.

4 Experiments

Datasets. Three in-domain datasets are used for training, and both in-domain and three out-of-domain datasets are used for validation of all models. Details are provide in Table 1 and 2. Available out-of-domain training data was not used due to experiments with transfer learning resulting in performance drops.

Evaluation Method. All experiments use the Exact-Match (EM) and F1 scores on the Dev datasets to evaluate each model in the same approach as the SQuAD paper [10].

[2]Package details: `https://github.com/UKPLab/sentence-transformers`

Dataset	Train	Dev	Test
SQuAD [10]	50,000	10,507	-
NewsQA [11]	50,000	4,212	-
NaturalQuestions [12]	50,000	12,836	-

Table 1: In-Domain Datasets

Dataset	Train	Dev	Test
DuoRC [13]	127	126	1248
RACE [14]	127	128	419
RelationExtraction [15]	127	128	2,693

Table 2: Out-of-Domain Datasets

Model	Gradient Steps	# Experts	SQuAD EM/F1	NewsQA EM/F1	NaturalQuestions EM/F1	Average EM/F1
BASELINE	45k	-	**63.2/77.4**	40.0/**57.9**	**53.3/70.0**	**55.1/71.0**
MoE-BASE	30k	6	62.6/76.8	39.0/57.2	51.0/67.8	53.6/69.6
MoE-Domain	20k	3	62.5/76.9	38.8/57.3	51.6/68.3	53.8/69.9
MoE-Adv	25k	3	62.3/76.7	40.1/57.6	52.0/68.3	54.1/69.9
Adversarial	45k	-	62.4/77.0	**41.1**/56.9	52.4/69.5	54.5/70.7

Table 3: In-Domain Results on Dev set. Best performance models for each dataset are in bold.

Model	Gradient Steps	# Experts	DuoRC EM/F1	RACE EM/F1	RelationExtraction EM/F1	Average EM/F1
BASELINE	45k	-	29.4/38.6	**28.1/40.0**	42.2/66.5	**33.3**/48.4
MoE-BASE	30k	6	34.1/42.8	23.4/36.4	41.4/67.1	33.0/**48.8**
MoE-Domain	20k	3	34.1/42.6	18.8/32.1	40.6/67.5	31.2/47.4
MoE-Adv	25k	3	**34.9/43.4**	20.3/34.6	43.0/67.5	32.7/48.6
Adversarial	45k	-	31.0/40.5	19.5/33.5	**47.7/71.3**	32.7/48.5

Table 4: Out-of-Domain Results on Dev set. Best performance models for each dataset are in bold.

Experiment Details. All models use a learning rate of 3e-5 and batch size of 16. MoE architectures have a hidden size of 1024 for all linear layers, The GRU in *MoE-Base* has 512 hidden units. For respective MoE models, $\lambda_{importance}$ is 0.1 as suggested by Takahashi et al., 2019 [2] and λ_{adv} is 1e-2 as suggested by Lee et al., 2019 [4]. The Adversarial model uses a discriminator with 3 linear layers with hidden sizes of 768.

Experiment Results. Table 3 and Table 4 display the results for all models. The best model on the out-of-domain dev datasets by average F1 score is the *MoE-Base* model. The increase in performance is a lower percentage than seen in Takahashi et a., 2019 [2]. As the implementation here is simpler, one potential cause may be that the natural language inference (NLI) subtask may cause the additional increase. This is left for further research.

MoE-Base is not the best model for any individual dataset. For out-of-domain datasets, DuoRC receives the best scores with *MoE-Adv*, and Relation Extraction is best with the Adversarial baseline. All models perform worse than the baseline on the RACE dataset.

The *MoE-Base* model is used for submission to assess performance on the test set. The results scores are a F1 of **57.382** and EM of **38.784**.

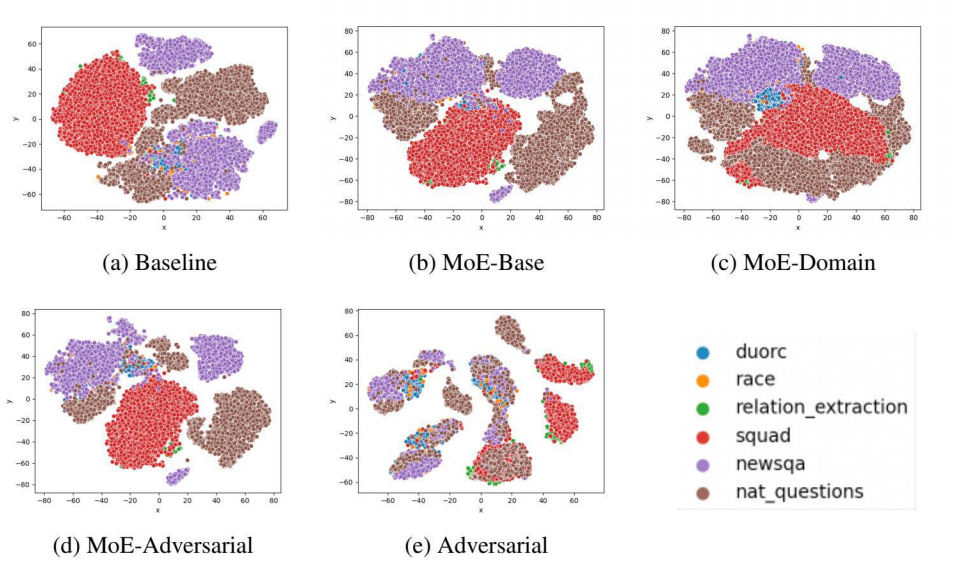
5 Analysis

In this section, two aspects are analyzed to determine each model's ability to (1) classify domains and (2) reason beyond simple heuristics and language biases.

5.1 Domain Identification

Mixture-of-experts and adversarial follow two different theories in order to learn generalized reading comprehension. Mixture-of-experts attempts to learn QA on individual domains. The idea is that then the out-of-domain data can be predicted using a linear combination of the experts from the in-domain data. Conversely, adversarial models attempt to learn domain indistinguishable hidden representations.

In order to visualize how these models represent examples from each domain, t-SNE plots can be used to view the representation space in 2D space (Ma et al., 2019 [3]). The output of the last DistilBERT



(a) Baseline (b) MoE-Base (c) MoE-Domain

(d) MoE-Adversarial (e) Adversarial

Figure 4: tSNE plots of the [CLS] token representations of the last transformer block of each model.

transformer is used to view the hidden representation of the [CLS] token. The tSNE embedding computations are computed using the sklearn package [3] and use default parameters. The resulting plots are included in Figure 4. For in-domain datasets, 5000 random examples from each source where used to generate the plots, and all examples from the out-of-domain dev sets were used.

Additionally, the average expert probabilities output by the gating networks of the gating network can be analyzed. The results are shown in Table 5.

From these results, there is clear distinction between hidden representations from each dataset in Figures 4a- 4d. These plots align to the expert probabilities seen in Table 5. An observation can be made that Relation Extraction is represented very similarly to SQuAD in all models. The single model where representation overlap is observed is the *Adversarial* model. The most overlap is seen in Natural Questions, NewsQA, DuoRC, and RACE. There are far more clusters observed, but these are not dataset specific. Future research into these cluster may provide insight into other aspects of the data examples.

5.2 Bias Analysis

Each dataset contains some level of bias examples that simple heuristics can solve that models may exploit. Table 6 shows the number of examples that contain each bias for individual datasets. The training set is used for in-domain datasets, and the dev set for out-of-domain. Notable differences occur in the proportion of lexical overlap examples within SQuAD, Natural Questions, and Relation Extraction. These are also the datasets with the highest performance. An open question is to see if this higher performance is due to models exploiting lexical overlap bias.

Lexical overlap occurs when the answer appears in most similar context sentence to the question. Using Figure 1 from Section 3.4 as an example, the context sentence that contains the answer is likely the only sentence needed for a human to answer the question. Therefore, lexical overlap might be viewed as examples that require a lower level of reasoning to solve, rather than a certain bias.

Figure 5 compares EM scores for all bias categories. Figures 5a- 5c show the out-of-domain datasets across each model implemented. The bias heuristics perform best on the relation extraction dataset, while RACE performs the worst. *MoE-Base* is the worst performing model on average across the bias heuristics, meaning it relies on heuristics the least. While the *Adversarial* model appears to have learned to perform best on the bias heuristics.

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`

Model	Dataset	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
MoE-Base	NaturalQuestions	0.158	0.158	0.172	**0.186**	0.167	0.158
	NewsQA	0.178	0.158	**0.184**	0.133	0.170	0.176
	SQuAD	0.112	**0.288**	0.134	0.195	0.120	0.151
	DuoRC	0.164	0.167	0.156	0.105	0.198	**0.210**
	RACE	0.174	**0.179**	**0.179**	0.167	0.161	0.140
	RelationExtraction	0.100	**0.346**	0.130	0.168	0.110	0.145
MoE-Domain	NaturalQuestions	0	0	**>0.999**	-	-	-
	NewsQA	0.003	**0.996**	0	-	-	-
	SQuAD	**0.998**	0.002	0	-	-	-
	DuoRC	0.239	**0.497**	0.246	-	-	-
	RACE	0.165	**0.835**	0	-	-	-
	RelationExtraction	**>0.999**	0	0	-	-	-
MoE-Adv	NaturalQuestions	0.331	**0.381**	0.288	-	-	-
	NewsQA	0.331	**0.371**	0.298	-	-	-
	SQuAD	0.340	0.221	0.**439**	-	-	-
	DuoRC	0.323	0.308	**0.369**	-	-	-
	RACE	0.323	0.321	**0.355**	-	-	-
	RelationExtraction	0.350	0.267	**0.383**	-	-	-

Table 5: Expert weights output by mixture-of-expert models. Highest weighted experts are shown in bold.

Dataset	Lexical Overlap	Interrogative Adverb	Empty Question
NaturalQuestions	26,303	46,071	50,000
NewsQA	8,664	48,703	50,000
SQuAD	30,978	49,155	50,000
DuoRC	41	126	126
RACE	27	123	128
RelationExtraction	100	128	128

Table 6: Summary of number of examples that are candidates for each bias. In-domain datasets uses the training data for analysis, while out-of-domain used the dev set.

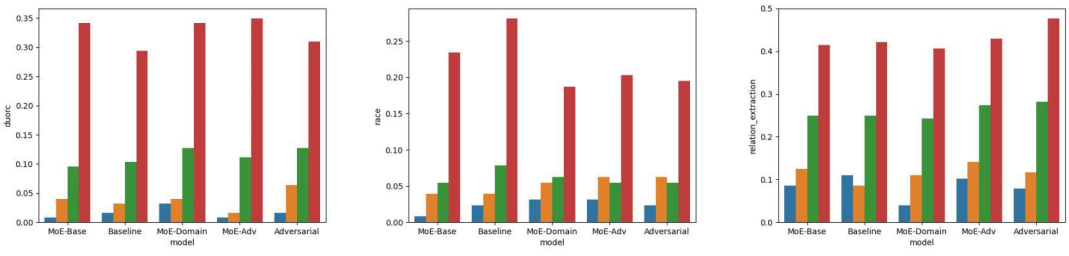
For in-domain datasets, further analysis is performed on *MoE-Base* to determine which datasets bias was learned from the most. Figure 5d shows a clear performance difference across all bias heuristics for Natural Questions and for lexical overlap on SQuAD. NewsQA, the worst performing dataset when using full data, sees the highest drops when using bias heuristics.

From this analysis, we can reason that certain datasets contain easier examples from a reading comprehension perspective. Using Natural Questions as an example, we can see that almost 40-50% of the EM score can be achieved by any of the bias types. Conversely, NewsQA receives much lower scores and thus requires a much higher level of reasoning.

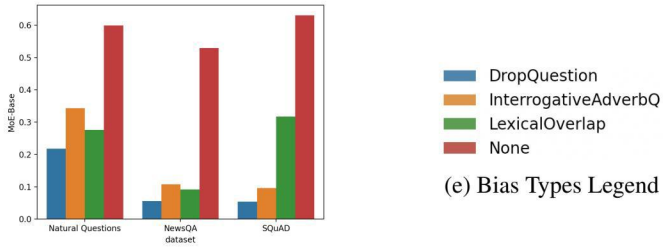
Debaising by Data Selection. Utama et al., 2020 [8] concluded that filtering out examples of bias from the training set will decrease overall performance of the model. This conclusion was confirmed with these models as well. The performance of *MoE-Base* using only unbiased examples dropped to EM and F1 scores of 43.78 and 26.18, respectively. These are drops of over 5 points each.

6 Conclusion

In this project, several models have been tested to build robust QA systems that can generalize to out-of-domain data. Based on the results, a mixture-of-experts model can achieve an F1 score 0.5 points above the baseline model on the dev datasets. Robust QA systems must be able to use in-domain datasets to learn to use a context and question to predict an answer. Two ways of viewing this are presented by analyzing domain representations and bias heuristics. Wu et al., 2020 [6] show that student-teacher model can utilize types of bias to decrease the influence of bias examples on



(a) DuoRC (b) RACE (c) Relation Extraction



(d) In-Domain - MoE-Base Results

(e) Bias Types Legend

Figure 5: Figures show EM scores performance when only using biased examples as input vs the overall model performance. The "None" bias type represents the model scores using all the entire dataset. Note that this comparison assumes non-biased examples receive an EM score of 0.

model training. Future work in this area could look at further bias heuristics to analyze more types of questions and levels of difficulty in datasets to improve student-teacher models.

References

[1] Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. In *Association for Computational Linguistics (ACL)*, 2018.

[2] Takumi Takahashi, Motoki Tangiguchi, Tomoki Taniguchi, and Tomoko Ohkuma. Cler: Cross-task learning with expert representation to generalize reading and understanding. In *Association for Computational Linguistics (ACL)*, 2019.

[3] Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Domain adaptation with bert-based domain classification and data selection. In *Association for Computational Linguistics (ACL)*, 2019.

[4] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. In *Association for Computational Linguistics (ACL)*, 2019.

[5] S. Sugawara, P. Stenetorp, K. Inui, and A. Aizawa. Assessing the benchmarking capacity of machine reading comprehension datasets. In *AAAI Conference on Artificial Intelligence*, 2020.

[6] Mingzhu Wu, Nafise Sadat Moosavi, Andreas Ruckle, and Irena Gurevych. Improving qa generalization by concurrent modeling of multiple biases. In *Association for Computational Linguistics (ACL)*, 2020.

[7] R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, 2019.

[8] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. In *Association for Computational Linguistics (ACL)*, 2020.

[9] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. In *Computational Natural Language Learning (CoNLL)*, 2017.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *CoRR, abs/1606.05250*, 2016.

[11] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Association for Computational Linguistics (ACL)*, 2017.

[12] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. In *Association for Computational Linguistics (ACL)*, 2019.

[13] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *Association for Computational Linguistics (ACL)*, 2018.

[14] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[15] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *arXiv preprint arXiv:1706.04115*, 2017.