Improved Robustness in Question-Answering via Multiple Techniques

Stanford CS224N Default Project

Jiankai Xiao
Department of Materials Science and Engineering
Stanford University
jkxiao@stanford.edu

Kexin Ding
Institute for Computational & Mathematical Engineering
Stanford University
cocodkx@stanford.edu

Abstract

As the transferability becomes a general issue in natural language processing (NLP), a question answering (QA) model that generalizes well on any domain is increasingly desirable. With the baseline model DistilBERT, we apply several techniques, including adversarial training, data augmentation, and task-adaptive pretraining, to improve the model performance on out-of-domain dataset. In this paper, task-adaptive pretraining gives the best performance, which has F1 score of 52.23 (an increase of 2.46) and EM score of 38.74 (an increase of 3.92). In additional, we also investigate the gain in robustness via each technique and whether the effect is complementary. The prospect and limitation of a combined system is discussed, and some future work to address the existing issues is also proposed.

1 Key Information to include

- Mentor: None
- External Collaborators (if you have any): N/A
- Sharing project: None

2 Introduction

Over the last few years, we have seen tremendous progress on fundamental natural language understanding problems. At the same time, there is increasing evidence that models fail to generalize beyond the training distribution. As the transferability becomes a general issue in natural language processing (NLP), a robust question answering (QA) system that generalizes well on any domain is increasingly desirable for the real world applications. However, for most of the current QA models, additional data are required to generalize to new domains, and these systems tend to overfit on specific domains. Hence, it is crucial to build domain-agnostic QA model that can learn domain-invariant features instead of focusing on the specific features. There are several existing methodologies that can improve the performance on QA task. The usage of pretrained language models, such as ELMo, GPT, BERT, etc., enables knowledge gained from pretraining to be transfered to the new model. Although some models might outperform human performance, but when it comes to out-of-domain data, they have poor performance, hence the idea of domain generalization is proposed. With methods such as using the most related in-domain datasets to test the unseen target domain data, using only domain-agnostic parameters to predict on out-of-domain datasets, or the meta-learning framework, the model performance on the unseen domains can be efficiently improved. For this project, we mainly investigate four methodologies: adversarial training[1], easy data augmentation[2], back-translation[3], and task-adaptive pretraining[4].

3 Related Work

There are several existing techniques that are relevant to improving the performance of the QA model on out-of-domain dataset.

In *Domain-agnostic Question-Answering with Adversarial Training*[1], they propose a method to regularize the QA model such that it could learn domain-invariant features. The model contains two components: a QA model and a domain discriminator. During training, the discriminator is trained to correctly predict the domain label of hidden representation from the model, while the model attempts to fool the discriminator so that the domain label of hidden representation becomes indistinguishable to the discriminator. The model and the discriminator work together to improve the model performance.

The technique proposed by *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*[2] has improved performance on five different text classification tasks. They define four operations: Synonym Replacement - randomly choose n words from the sentence and replace with their synonyms; Random Insertion - insert the synonym to a random word in the sentence at a random position; Random Swap - randomly swap two words in the sentence for n times; Random Deletion - randomly delete each word with a probability. For small datasets, they augment data by randomly performing one of the four operations on each sentence. This EDA technique substantially boosts the model performance and reduces overfitting with small training dataset.

In *An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering*[3], they utilize the back-translation methodology to perform data augmentation on the training dataset. For each query and context pair (q, c), they use the back-translation model to generate both a query paraphrase q' and a context paraphrase c'. Then, with probability $P_q(x)$, they create a new pair that includes the paraphrase q' instead of q, and independently, choose the paraphrase c' over c with probability $P_c(x)$. When either q' or c' is sampled, add the augmented example to the training data. With this sampling strategy, they could control the frequency of including the query or text augmentation.

In *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*[4], they mainly focus on two variations for adapting pretrained neural language models to domains and tasks. The domain-adaptive pretraining (DAPT) approach further pretrains the pretrained model on a large corpus of unlabeled domain-specific data, and the task-adaptive pretraining (TAPT) refers to pretraining on the unlabeled training set for a given task. Their findings suggest that it might be valuable to complement work on ever-larger language models with parallel efforts to identify and use domain and task relevant corpora to specialize models.

4 Approach

4.1 Baseline

The baseline system finetunes the pretrained DistilBERT[5] model ('distilbert-base-uncased') for QA task on in-domain and out-of-domain dataset.

4.2 Adversarial Training

As proposed by [1], the domain discriminator $\mathcal{D}$ minimizes the cross-entropy loss (where l is the domain label, $\mathbf{h} \in \mathbb{R}^d$ is the hidden representation, K is the number of domain classes, N_k is the number of data in domain k and $N = \sum_{k=1}^{K} N_k$ is the total number of data):

$$\mathcal{L}_{\mathcal{D}} = -\frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \log P_\phi(l_i^{(k)} | \mathbf{h}_i^{(k)})$$

The QA model first minimizes the negative log-likelihood of the start and end position of answer for the standard QA task. In addition, the QA model maximizes the cross-entropy loss of $P_\phi(l_i^{(k)}|\mathbf{h}_i^{(k)})$, which is to minimize the KL divergence between the uniform distribution over K domain classes and the prediction of discriminator (where $\mathcal{U}(l)$ is the uniform distribution):

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k} KL\left(\mathcal{U}(l)||P_\phi(l_i^{(k)}|\mathbf{h}_i^{(k)})\right)$$

Then the total loss for the QA model to minimize is (where λ is a hyper-parameter indicating the importance of $\mathcal{L}_{adv}$)

$$\mathcal{L}_{QA} + \lambda\mathcal{L}_{adv}$$

For this technique, We refer to [1]. We adopt the same discriminator architecture (3 layers of fully-connected neural network with hidden size of 768), and re-implement the adversarial system and the training logic in order to be compatible with the interface of DistilBERT model.

4.3 Data Augmentation

As proposed by [2], Easy Data Augmentation (EDA) contains 4 techniques: random replacement (RR), random insertion (RI), random swap (RS) and random deletion (RD). Since the original technique is designed for the text classification task, a few modifications are made to adapt the system to QA task. Our replacement/insertion algorithm queries synonyms from NLTK 'wordnet' excluding stopwords in NLTK 'stopwords'. Our insertion/deletion algorithm also considers the effect of punctuation to introduce more variants of a sentence. Our swap algorithm randomly samples n words in a sentence and shuffles them, which is different from the original implementation. Considering the sanity of answer in the QA task, our insertion/swap/deletion algorithm preserves the answer in the context (the text from start position to end position). The number of words to be replaced/inserted/swapped/deleted, n, is determined by the length of a sentence (excluding answer if it exists) multiplied by the percent of words to be changed (represented by n). For each pair of question and context (q, c), our algorithm creates a new pair that augments the question to be q' with the fixed probability (represented by P) and the context to be c' with the same probability but independently, and then if either one is sampled, the new pair is included into the training set. For this technique, we refer to [2], but the implementation is mostly original.

We have also implemented back-translation using Google Translate API with German being the default pivot language. Considering the sanity of answer in the QA task, we have several solutions. First, we can skip the sentence in which the answer exists. Second, we can back-translate both the context and the answer text, and programmatically search for the exact match of answer text in the context. Third, instead of exact match, we can search for the most similar text using NLP model. The first and the second algorithm have been implemented. For this technique, we do not refer to any external repository. Nevertheless, due to the long time it takes to send HTTP request and the time constraint of this project, we are unable to apply this technique.

4.4 Task-Adaptive Pretraining

As proposed by [4], before finetuning on QA data, a second-phase masked language modeling (MLM) pretraining on unlabeled data from new domains can familiarize the model with domain-specific vocabulary and sentence structure. The pairs of question and context are masked for MLM task using the same strategy as BERT (15% randomly sampled tokens are masked for prediction, among which 80% are replaced by [MASK], 10% are replaced by random tokens and 10% remain unchanged). To emphasize the importance of out-of-domain data, the out-of-domain dataset is artificially enlarged by repeating each pair 100 times. For this technique, we refer to [3] and re-implement the pretraining logic in order to be compatible with the interface of DistilBERT model.

[1] https://github.com/seanie12/mrqa

[2] https://github.com/jasonwei20/eda_nlp

[3] https://github.com/allenai/dont-stop-pretraining

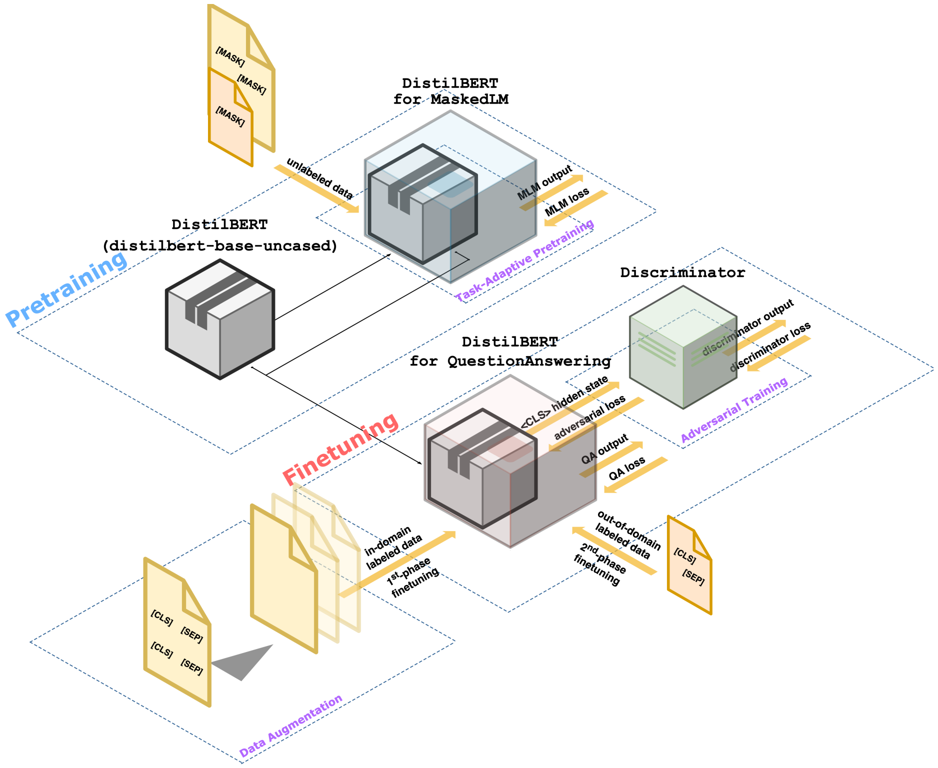


Figure 1: model architecture

4.5 Model Architecture

With abovementioned techniques implemented, our model architecture is shown in Figure 1. The black lines show how the DistilBERT model is loaded and the yellow lines show the flow of data or loss during training. The region labeled ‘pretraining’ encompass all the processes that are not task-specific where the system is trained with MLM task, while the region labeled ‘finetuning’ encompass all the processes that are QA-task-specific where the system is trained with QA task. The regions with purple labels are techniques as described above. The task-adaptive pretraining (TAPT) process also refers to the second phase of pretraining. Furthermore, finetuning on in-domain dataset refers to the first phase of finetuning and finetuning on out-of-domain dataset refers to the second phase of finetuning.

We set a feature flag for each technique and investigated the gain in robustness via each technique and possible combination of techniques. If the ‘TAPT’ flag is on, the QA model loads DistilBERT from the final state of TAPT, otherwise, it loads from ‘distilbert-base-uncased’. If the ‘EDA’ flag is on, the in-domain dataset (train set) is first augmented by EDA with a configuration file during each epoch and then used for training, otherwise, the original in-domain dataset is used. If the ‘adversarial’ flag is on, the QA model and the discriminator are trained together and both the QA loss and the adversarial loss are included (this training process refers to adversarial training), otherwise, only the QA model is trained and only the QA loss is included (this training process refers to baseline training).

5 Experiments

5.1 Data

We use the in-domain datasets SQuAD[6], Natural Questions[7], and NewsQA[8] and out-of-domain datasets composed by unseen question answering datasets DuoRC[9], RACE[10], and RelationExtraction[11] provided and preprocessed by the course teaching team. The in-domain datasets are splited into train and validation sets, and the out-of-domain datasets are splited into test, train, and validation sets. The training and validation sets are composed by triples of context, question, and answer, where the answer is a span from the context. And the test set is composed by only context and question. We use the in-domain training dataset to train the baseline model and use the in-domain validation dataset to evaluate the training result. The out-of-domain training dataset is used for additional finetuning, and the out-of-domain validation dataset is used for checking the model performance and for setting hyperparameters. Finally, we use the out-of-domain test dataset to evaluate performance of the model after applying the techniques.

5.2 Evaluation method

The performance of our QA system is measured via Exact Match (EM) score and F1 score. EM is a binary measure of whether the output matches the ground truth exactly. F1 is the harmonic mean of precision and recall. If multiple ground truths are provided, the maximum EM and F1 score are recorded. The EM and F1 score are averaged across the entire evaluation dataset and finally reported.

To investigate whether the gain in robustness via each technique is complimentary, we refer the set of data that have F1 larger than or equal to 50 as the correctly predicted set, and visualize the result with Venn diagram.

5.3 Experimental details

For the baseline, during the first phase, the system is trained on in-domain dataset for 3 epochs (each epoch takes about 5.5 hours on Microsoft Azure Standard NC6) and evaluated each 5000 steps. During the second phase, it is then trained on out-of-domain dataset for 10 epochs from the checkpoint with the highest F1 score and evaluated each 10 steps. The batch size is 16 and the learning rate is $3e-5$. The other experiments follow the same procedure as the baseline.

For adversarial training, the hyperparameter λ is tuned with multiple experiments as shown in Table 1, each epoch takes about 8 hours on Microsoft Azure Standard NC6). For EDA, first, only random replacement is included and two hyperparameters, percent ($\%$) and probability (P), are tuned as shown in Table 2 and Table 3 respectively. Then random insertion/swap/deletion are all included and the hyperparameter $\%$ is tuned as shown in Table 4. For TAPT, DistilBERT is pretrained on in-domain and out-of-domain (repeated 100 times) masked dataset for 1 epoch (which takes about 6 hours on Microsoft Azure Standard NC6).

The complete experiment combining adversarial training, EDA, and TAPT with separately tuned hyperparameters is then conducted (each epoch takes 3.3 hours on Microsoft Azure Standard NC6s_v2). The combination of adversarial training and EDA and of EDA and TAPT are also performed.

5.4 Results

5.4.1 Hyperparameters Tuning

The tuned hyperparameters with the highest F1 score is as below: $\lambda = 0.1$ for adversarial training, only random replacement with $\%$ = 0.2, P = 0.1 for EDA. All the results in this section are evaluated on the out-of-domain dev set.

Table 1: F1 and EM scores with adversarial training (different λ)

Model	**F1**	**EM**
baseline	49.77	34.82
adversarial ($\lambda = 0.01$)	49.82	35.08
adversarial ($\lambda = 0.02$)	49.78	35.60
adversarial ($\lambda = 0.05$)	49.72	33.77
adversarial ($\lambda = 0.1$)	50.25	34.55
adversarial ($\lambda = 0.2$)	47.79	34.03
adversarial ($\lambda = 0.5$)	47.16	31.94

Table 2: random replacement (different percent)

Model	**F1**	**EM**
baseline	49.77	34.82
EDA-RR (% = 0.1, P = 0.1)	48.17	31.41
EDA-RR (% = 0.2, P = 0.1)	50.31	34.82
EDA-RR (% = 0.3, P = 0.1)	49.44	33.77

Table 3: random replacement (different probability)

Model	**F1**	**EM**
baseline	49.77	34.82
EDA-RR (% = 0.2, P = 0.1)	50.31	34.82
EDA-RR (% = 0.2, P = 0.2)	49.22	33.77

Table 4: random insertion + random swap + random deletion

Model	**F1**	**EM**
baseline	49.77	34.82
EDA-RR (% = 0.2, P = 0.1)	50.31	34.82
EDA-RR (% = 0.2, P = 0.1) + others (% = 0.1, P = 0.1)	48.80	34.82
EDA-RR (% = 0.2, P = 0.1) + others (% = 0.2, P = 0.1)	48.67	34.03

Table 5: task-adaptive pretraining

Model	**F1**	**EM**
baseline	49.77	34.82
TAPT	52.23	38.74

Table 6: combined experiments

Model	**F1**	**EM**
baseline	49.77	34.82
adversarial + EDA-RR + TAPT	50.50	36.13
adversarial + EDA-RR	51.01	35.03
EDA-RR + TAPT	50.30	34.82

5.4.2 Test Set Results

As shown the test leaderboard of RobustQA track, our system has achieved F1 score of 58.723 and EM score of 40.917.

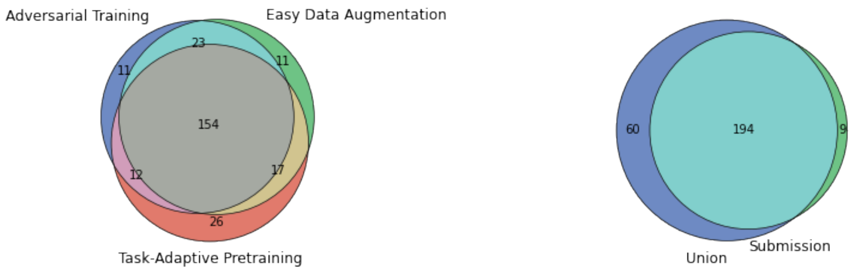
6 Analysis

The optimal value of λ is 0.1 in this paper, which is different from the value of 0.01 in [1]. This might result from the smaller size of dataset for training and the fewer domain classes for adversarial training, which indicates that the importance of adversarial loss should be increased if only a smaller dataset is available. Moreover, with only 3 domain classes for the discriminator to predict, the performance of discriminator might degrade and become unstable. A possible solution is to train a clustering model on the hidden state representation of transformer and the cluster label can act as the target for the discriminator to predict.

The best strategy for EDA in this paper is only using random replacement, while random insertion/swap/deletion are proved to degrade the performance. Since a QA system sometimes relies on some key word or specific sentence structure to predict the answer, random insertion/swap/deletion will also damage this kind of knowledge and confuse the system during training, while random replacement can preserve these features but still break the brittle correlation among the training data.

To improve the performance and stability of a QA system, the second phase of finetuning on out-of-domain dataset is important. For experiments with either only adversarial training or only EDA, the system achieves the highest F1 within 3 epochs during the second phase of finetuning, which indicates that the system is overfitted on in-domain dataset and does not have much potential to achieve better robustness on out-of-domain dataset. For experiments with a combination of multiple techniques, the system achieves the highest F1 within 10 epochs, which indicates that the system is less overfitted and might have learned some domain-invariant knowledge. Especially for experiments with only TAPT, the system can still achieve higher F1 after 10 epochs with a smaller learning rate, which implies that the system might be still underfitted after 10 epochs.

To understand the effect of TAPT, we compare the results as shown in Table 6. The second experiment (adversarial + EDA-RR + TAPT) has F1 of 50.50 while the third experiment (adversarial + EDA-RR) has higher F1 of 51.01, which means that TAPT plays a negative role in the combination of multiple techniques. This is because either adversarial training or data augmentation attempts to learn some domain-invariant knowledge so that the system generalizes well on any new domain, while the second phase of pretraining is domain-specific which introduces an adverse effect to the whole system.



(a) three correctly predicted sets

(b) the union of sets and the set of combined system

Figure 2: Venn diagrams

As multiple techniques are combined, the comprehensive system does not achieved a significantly improved F1, which means that the gain in robustness does not increase as expected. The adversarial effect between domain-invariant and domain specific knowledge has been discussed above. However, another aspect to investigate is whether the gain via each technique is indeed complementary. To visualize this, the correctly predicted sets for adversarial training, EDA and TAPT are extracted, as shown in 2a. These three sets are not completely overlapped, which implies that the combined system has the potential to achieve higher F1. Naively, we might expect the combined system to correctly predict the union of all sets, as shown in 2b, but it fails on a large portion of the union. This also suggests that the Mixture-of-Experts technique might possibly solve the issue.

7 Conclusion

In this paper, we experiment and combine several techniques to build a QA model that generalizes well on the out-of-domain datasets.

With the baseline model DistilBERT[5], we first separately apply adversarial training[1], easy data augmentation[2], and task-adaptive pretraining[4] and tune the important hyperparameters for each technique. Then, we combine several techniques to further improve the system, and evaluate the performance with F1 and EM score.

With the task-adaptive pretraining technique, we eventually achieve F1 score of 52.23 (+2.46 compared to the baseline) and EM score of 38.74 (+3.92 compared to the baseline). One the test set, the combination of adversarial training and easy data augmentation achieves F1 score of 58.723 and EM score of 40.917.

We discuss the increased value of λ in adversarial training and why it might be important for a smaller dataset. For easy data augmentation, random replacement is found to be most effective since it preserves knowledge-related features but still breaks the brittle correlation among the training data. We also emphasize the importance of the second phase of finetuning, which can improve the performance and stability of the QA system. We further investigate the domain-invariant approach and the domain-specific approach, both of which can generalize to out-of-domain data, and the adversarial effect between them in a combined system. With the visualization of correctly predict set of each technique, we study the issue of complementary gain in robustness and propose that the Mixture-of-Experts technique might be a possible solution.

References

[1] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 196–202. Association for Computational Linguistics, November 2019.

[2] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics.

[3] Longpre Shayne, Lu Yi, Tu Zhucheng, and DuBois Chris. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 220–227. Association for Computational Linguistics, December 2019.

[4] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Workshop on Energy Efficient Machine Learning and Cognitive Computing (5th edition)*. Neural Information Processing Systems Foundation, October 2019.

[6] Rajpurkar Pranav, Zhang Jian, Lopyrev Konstantin, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, June 2016.

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

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

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

[10] Lai Guokun, Xie Qizhe, Liu Hanxiao, Yang Yiming, and Hovy Eduard. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[11] Levy Omer, Seo Minjoon, Choi Eunsol, and Zettlemoyer Luke. Zero-shot relation extraction via reading comprehension. arXiv, 2017.